

# CA-IDMS®

---

Reports  
15.0



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

## **Second Edition, October 2001**

© 2001 Computer Associates International, Inc.  
All rights reserved.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

---

<b>How to Use This Manual</b> . . . . .	xiii
 <b>Chapter 1. Introduction</b> . . . . .	1-1
1.1 Reports stored as dictionary modules . . . . .	1-3
1.2 Dictionary reports . . . . .	1-4
1.3 Run-time reports . . . . .	1-5
1.4 Producing CA-IDMS reports . . . . .	1-6
1.4.1 DATABASE parameter . . . . .	1-6
1.4.2 PARAM= parameter . . . . .	1-7
1.4.3 REPORT= parameter . . . . .	1-9
1.4.4 KEY parameter . . . . .	1-10
1.4.5 SELECT and BYPASS parameters . . . . .	1-10
1.4.5.1 Examples . . . . .	1-13
1.5 Modifying CA-IDMS reports . . . . .	1-15
1.6 CA-CULPRIT Security Considerations . . . . .	1-16
 <b>Chapter 2. Standard Dictionary Reports — DREPORTS</b> . . . . .	2-1
2.1 Report on the dictionary . . . . .	2-3
2.2 Information obtained from the DDLDML area . . . . .	2-4
2.3 Uses for dictionary reports . . . . .	2-5
2.4 Dictionary report categories . . . . .	2-6
2.5 Types of reports . . . . .	2-7
2.6 Producing dictionary reports . . . . .	2-8
2.6.1 Syntax . . . . .	2-8
2.6.2 Parameters . . . . .	2-8
2.6.3 Key field names for key reports . . . . .	2-9
2.6.4 Examples . . . . .	2-10
2.7 Basic entity reports . . . . .	2-12
2.7.1 System reports (DREPORTs 003, 053, 073) . . . . .	2-12
2.7.1.1 System Summary Report (DREPORT 053) . . . . .	2-13
2.7.1.2 System Detail and Key Reports (DREPORTs 003 and 073) . . . . .	2-13
2.7.2 User reports (DREPORTs 004, 054, 074) . . . . .	2-15
2.7.2.1 User Summary Report (DREPORT 054) . . . . .	2-15
2.7.2.2 User Detail and Key Reports (DREPORTs 004 and 074) . . . . .	2-16
2.7.3 Program reports (DREPORTs 005, 055, 075) . . . . .	2-20
2.7.3.1 Program Summary Report (DREPORT 055) . . . . .	2-20
2.7.3.2 Program Detail and Key Reports (DREPORTs 005 and 075) . . . . .	2-20
2.7.4 Module reports (DREPORTs 006, 056, 076) . . . . .	2-23
2.7.4.1 Module Summary Report (DREPORT 056) . . . . .	2-23
2.7.4.2 Module Detail and Key Reports (DREPORTs 006 and 076) . . . . .	2-24
2.7.5 File reports (DREPORTs 007, 057, 077) . . . . .	2-26
2.7.5.1 File Summary Report (DREPORT 057) . . . . .	2-26
2.7.5.2 File Detail and Key Reports (DREPORTs 007 and 077) . . . . .	2-26
2.7.6 Record reports (DREPORTs 008, 038, 058, 078) . . . . .	2-29
2.7.6.1 Record Summary Report (DREPORT 058) . . . . .	2-29
2.7.6.2 Record Detail and Key Report (DREPORTs 008 and 078) . . . . .	2-29
2.7.6.3 Record/Attribute Key Report (DREPORT 038) . . . . .	2-32

2.7.7	Element reports (DREPORTs 009, 010, 039, 059, 079)	2-33
2.7.7.1	Element Summary Report (DREPORT 059)	2-34
2.7.7.2	Element Detail and Key Reports (DREPORTs 009 and 079)	2-34
2.7.7.3	Inactive Element Detail Report (DREPORT 010)	2-36
2.7.7.4	Attribute/Element Key Report (DREPORT 039)	2-37
2.8	Site-specific entity reports	2-39
2.8.1	Class Reports (DREPORTs 001 and 071)	2-39
2.8.2	Attribute reports (DREPORTs 002 and 072)	2-41
2.8.3	User-defined entity reports (DREPORTs 019 and 089)	2-43
2.9	Teleprocessing entity reports	2-45
2.9.1	Task reports (DREPORTs 011, 061, 081)	2-46
2.9.1.1	Task Summary Report (DREPORT 061)	2-46
2.9.1.2	Task Detail and Key Reports (DREPORTs 011 and 081)	2-46
2.9.2	Queue reports (DREPORTs 012, 062, 082)	2-48
2.9.2.1	Queue Summary Report (DREPORT 062)	2-48
2.9.2.2	Queue Detail and Key Reports (DREPORTs 012 and 082)	2-48
2.9.3	Destination reports (DREPORTs 013, 063, 083)	2-50
2.9.3.1	Destination Summary Report (DREPORT 063)	2-50
2.9.3.2	Destination Detail and Key Reports (DREPORTs 013 and 083)	2-50
2.9.4	Logical terminal reports (DREPORTs 014, 064, 084)	2-52
2.9.4.1	Logical Terminal Summary Report (DREPORT 064)	2-52
2.9.4.2	Logical Terminal Detail and Key Reports (DREPORTs 014 and 084)	2-52
2.9.5	Physical terminal reports (DREPORTs 015, 065, 085)	2-54
2.9.5.1	Physical Terminal Summary Report (DREPORT 065)	2-54
2.9.5.2	Physical Terminal Detail and Key Reports (DREPORTs 015 and 085)	2-55
2.9.6	Line reports (DREPORTs 016, 066, 086)	2-56
2.9.6.1	Line Summary Report (DREPORT 066)	2-56
2.9.6.2	Line Detail and Key Reports (DREPORTs 016 and 086)	2-57
2.9.7	Panel reports (DREPORTs 017, 067, 087)	2-58
2.9.7.1	Panel Summary Report (DREPORT 067)	2-58
2.9.7.2	Panel Detail and Key Reports (DREPORTs 017 and 087)	2-59
2.9.8	Map reports (DREPORTs 018, 068, 088)	2-60
2.9.8.1	Map Summary Report (DREPORT 068)	2-61
2.9.8.2	Map Detail and Key Reports (DREPORTs 018 and 088)	2-61
2.10	Cross-reference reports	2-64
2.10.1	File/Record Cross-Reference Report (DREPORT 020)	2-65
2.10.2	File Synonym Cross-Reference Report (DREPORT 021)	2-66
2.10.3	Record Synonym Cross-Reference Report (DREPORT 022)	2-66
2.10.4	Element Synonym Cross-Reference Report (DREPORT 023)	2-67
2.10.5	Element Description Report (DREPORT 024)	2-68
2.10.6	Element Designator Report (DREPORT 025)	2-70
2.10.7	File Activity Report (DREPORT 026)	2-70
2.10.8	IDMS Set Activity Report (DREPORT 027)	2-71
2.10.9	IDMS Record Activity Report (DREPORT 028)	2-72
2.10.10	IDMS Area Activity Report (DREPORT 029)	2-73
2.10.11	Element/Program Cross-Reference Report (DREPORT 030)	2-74
2.11	Special-purpose report modules	2-76
2.11.1	Comment/Nest Resolution Report Module (DREPORT 00)	2-76
2.11.2	Level Number Report (DREPORT 050)	2-76

2.11.3	Module Text to Card Utility (DREPORT 051)	2-77
2.11.4	Module Text to File Utility (DREPORT 052)	2-78
<b>Chapter 3.</b>	<b>CA-ICMS Catalog Reports — DREPORTS</b>	3-1
3.1	Purpose	3-3
3.2	Information stored in DDLDDL area of dictionary	3-4
3.3	Uses for catalog reports	3-5
3.4	Summary of catalog reports	3-6
3.5	Producing catalog reports	3-7
3.5.1	Syntax	3-7
3.5.2	Parameters	3-7
3.5.3	Examples	3-8
3.6	DREPORT 090 — Catalog Summary Report	3-9
3.6.1	Contents	3-9
3.6.2	Field descriptions	3-9
3.7	DREPORT 091 — Catalog Detail Report	3-11
3.7.1	Contents	3-11
3.7.2	Field descriptions	3-12
3.8	DREPORT 092 — Group Detail Report	3-14
3.8.1	Contents	3-14
3.8.2	Field descriptions	3-14
3.9	DREPORT 093 — User Detail Report	3-16
3.9.1	Contents	3-16
3.9.2	Field descriptions	3-16
3.10	DREPORT 094 — Folder Detail Report	3-19
3.10.1	Contents	3-19
3.10.2	Field descriptions	3-19
3.11	DREPORT 095 — Object Detail Report	3-21
3.11.1	Contents	3-21
3.11.2	Field descriptions	3-21
3.12	DREPORT 096 — Catalog Summary Key Report	3-23
3.12.1	Contents	3-23
3.12.2	Field descriptions	3-23
3.13	DREPORT 097 — Catalog Detail Key Report	3-24
3.13.1	Contents	3-24
3.13.2	Field descriptions	3-24
<b>Chapter 4.</b>	<b>DC/UCF System Reports — CREPORTS</b>	4-1
4.1	Overview	4-3
4.2	Uses for DC/UCF system reports	4-4
4.3	Summary of CREPORTs	4-5
4.4	Producing DC/UCF system reports	4-7
4.4.1	Syntax	4-8
4.4.2	Parameters	4-8
4.4.3	Examples	4-8
4.5	CA-ADS Parameter Reports (CREPORTs 040 and 045)	4-10
4.5.1	Contents	4-10
4.5.2	Field descriptions	4-10
4.6	Defined Devices Report (CREPORT 029)	4-13
4.6.1	Contents	4-13

4.6.2 Field descriptions . . . . .	4-13
4.7 Defined Messages Report (CREPORT 028) . . . . .	4-14
4.7.1 Contents . . . . .	4-14
4.7.2 Field descriptions . . . . .	4-14
4.8 Destination Reports (CREPORTs 007 and 024) . . . . .	4-16
4.8.1 Contents . . . . .	4-16
4.8.2 Field descriptions . . . . .	4-16
4.9 Load Area Report (CREPORT 050) . . . . .	4-18
4.9.1 Contents . . . . .	4-18
4.9.2 Field descriptions . . . . .	4-18
4.10 Mapping Reports (CREPORTs 030 through 035) . . . . .	4-19
4.10.1 Contents . . . . .	4-19
4.10.2 Field descriptions . . . . .	4-21
4.11 Nodes and Resource Table Reports (CREPORTs 043 and 044) . . . . .	4-23
4.11.1 Contents . . . . .	4-23
4.11.2 Field descriptions . . . . .	4-24
4.12 Module Text to Card Utility (CREPORT 051) . . . . .	4-25
4.13 Module Text to File Utility (CREPORT 052) . . . . .	4-27
4.14 Network Description Reports (CREPORTs 001-003, 014-018) . . . . .	4-29
4.14.1 Contents . . . . .	4-29
4.14.2 Sample reports . . . . .	4-29
4.14.3 Field descriptions . . . . .	4-31
4.15 CA-OLQ Reports (CREPORTs 041 and 046) . . . . .	4-34
4.15.1 Contents . . . . .	4-34
4.15.2 Field descriptions . . . . .	4-34
4.16 Program Description Reports (CREPORTs 004 and 019) . . . . .	4-37
4.16.1 Contents . . . . .	4-37
4.16.2 Field descriptions . . . . .	4-37
4.17 Queue Description Reports (CREPORTs 006, 022, and 023) . . . . .	4-40
4.17.1 Contents . . . . .	4-40
4.17.2 Field descriptions . . . . .	4-40
4.18 Symbol Table Report (CREPORT 053) . . . . .	4-42
4.18.1 Contents . . . . .	4-42
4.18.2 Field descriptions . . . . .	4-42
4.19 System Options Reports (CREPORTs 011 and 025) . . . . .	4-43
4.19.1 Contents . . . . .	4-43
4.19.2 Field descriptions . . . . .	4-44
4.20 Task Description Reports (CREPORTs 005, 020, and 021) . . . . .	4-53
4.20.1 Contents . . . . .	4-53
4.20.2 Field descriptions . . . . .	4-53
4.21 Builder codes . . . . .	4-57
 <b>Chapter 5. CA-ADS Reports — AREPORTS . . . . .</b>	 5-1
5.1 Overview . . . . .	5-3
5.2 Summary table . . . . .	5-4
5.3 Uses for CA-ADS reports . . . . .	5-5
5.4 Producing CA-ADS reports . . . . .	5-6
5.4.1 Syntax . . . . .	5-6
5.4.2 Parameters . . . . .	5-6
5.4.3 Examples . . . . .	5-7
5.5 AREPORT 001 and 002 — ADS Dialogs and Their Components . . . . .	5-8

5.5.1	Contents	5-8
5.5.2	Field descriptions	5-9
5.6	AREPORT 003 — ADS Dialogs by Process Key	5-12
5.6.1	Contents	5-12
5.6.2	Field descriptions	5-12
5.7	AREPORT 004 — ADS Dialogs by Record Key	5-14
5.7.1	Contents	5-14
5.7.2	Field descriptions	5-14
5.8	AREPORT 005 — ADS Dialogs by Subschema Key	5-16
5.8.1	Contents	5-16
5.8.2	Field descriptions	5-16
5.9	AREPORT 006 — ADS Dialogs by Map Key	5-17
5.9.1	Contents	5-17
5.9.2	Field descriptions	5-17
<b>Chapter 6.</b>	<b>CA-IDMS/DB SQL Dictionary Reports — QREPORTS</b>	6-1
6.1	Overview	6-3
6.2	Summary table	6-4
6.3	QREPORT 001 —SQL Column Name Report	6-5
6.3.1	Contents	6-5
6.3.2	Symbolic Parameter Overrides	6-5
6.3.3	Job Submission	6-5
6.3.4	Field descriptions	6-6
6.4	QREPORT 002 —Table and Column Report	6-8
6.4.1	Contents	6-8
6.4.2	Symbolic Parameter Overrides	6-8
6.4.3	Job Submission	6-8
6.4.4	Field descriptions	6-9
6.5	QREPORT 003 —Schema and Table Report	6-13
6.5.1	Contents	6-13
6.5.2	Symbolic Parameter Overrides	6-13
6.5.3	Job Submission	6-13
6.5.4	Field descriptions	6-14
6.6	QREPORT 004 —SQL Access Module Information	6-17
6.6.1	Contents	6-17
6.6.2	Symbolic Parameter Overrides	6-17
6.6.3	Job Submission	6-17
6.6.4	Field descriptions	6-18
6.7	QREPORT 005 —SQL Table Access Report	6-20
6.7.1	Contents	6-20
6.7.2	Symbolic Parameter Overrides	6-20
6.7.3	Job Submission	6-20
6.7.4	Field descriptions	6-21
6.8	QREPORT 006 —SQL Table Syntax Report	6-22
6.8.1	Contents	6-22
6.8.2	Symbolic Parameter Overrides	6-22
6.8.3	Job Submission	6-22
6.8.4	Field descriptions	6-24
6.9	QREPORT 007 —SQL Table Index Report	6-25
6.9.1	Contents	6-25

6.9.2	Symbolic Parameter Overrides	6-25
6.9.3	Job Submission	6-25
6.9.4	Field descriptions	6-26
6.10	QREPORT 008 —SQL Table Constraint Report	6-28
6.10.1	Contents	6-28
6.10.2	Symbolic Parameter Overrides	6-28
6.10.3	Job Submission	6-28
6.10.4	Field descriptions	6-29
<b>Chapter 7.</b>	<b>ASF Row-Level Security Reports — IREPORTS</b>	<b>7-1</b>
7.1	Overview	7-3
7.2	Producing ASF row-level security reports	7-4
7.2.1	Steps	7-4
7.2.2	Syntax	7-4
7.2.3	Parameters	7-4
7.2.4	Summary table	7-5
7.2.5	Example	7-5
7.3	ASF row-level security reports	7-6
<b>Chapter 8.</b>	<b>CA-IDMS/DB Journal Reports — JREPORTS</b>	<b>8-1</b>
8.1	Overview	8-3
8.2	Summary of journal reports	8-4
8.3	Uses for journal reports	8-5
8.4	Types of journal records	8-6
8.5	Summary of records required for journal reports	8-14
8.6	Producing journal reports	8-15
8.6.1	Uses archive journal file as input	8-15
8.6.2	Required CA-CULPRIT modules	8-15
8.6.3	Syntax	8-15
8.6.4	Parameters	8-16
8.6.5	Examples	8-17
8.6.6	Operating system considerations	8-18
8.7	JREPORT 001 — Transaction Summary	8-20
8.7.1	Purpose	8-20
8.7.2	Field descriptions	8-20
8.8	JREPORT 002 — Program Termination Statistics	8-22
8.8.1	Purpose	8-22
8.8.2	Field descriptions	8-23
8.9	JREPORT 003 — Program I/O Statistics	8-25
8.9.1	Purpose	8-25
8.9.2	Field descriptions	8-25
8.10	JREPORT 004 — Program Summary	8-27
8.10.1	Purpose	8-27
8.10.2	Field descriptions	8-27
8.11	JREPORT 005 — Detail Area/Transaction	8-30
8.11.1	Purpose	8-30
8.11.2	Field descriptions	8-30
8.12	JREPORT 006 — Detail Program/Area	8-32
8.12.1	Purpose	8-32
8.12.2	Field descriptions	8-32
8.13	JREPORT 007 — Area Summary	8-34



8.13.1 Purpose . . . . .	8-34
8.13.2 Field descriptions . . . . .	8-34
8.14 JREPORT 008 — Formatted Record Dump . . . . .	8-36
8.14.1 Purpose . . . . .	8-36
8.14.2 Field descriptions . . . . .	8-37
 <b>Chapter 9. DC/UCF Statistics Reports — SREPORTS</b> . . . . .	9-1
9.1 Overview . . . . .	9-3
9.2 Summary of statistics reports . . . . .	9-4
9.3 Uses for statistics reports . . . . .	9-5
9.4 Other tools available . . . . .	9-6
9.5 Types of statistics records . . . . .	9-7
9.5.1 Statistics record type . . . . .	9-7
9.5.2 Layout of statistics log records . . . . .	9-7
9.5.3 Task or transaction statistics . . . . .	9-8
9.5.4 Order of DSECTs for task and transaction statistics . . . . .	9-8
9.5.5 System statistics . . . . .	9-9
9.5.6 Order of DSECTs for system statistics . . . . .	9-9
9.5.7 Histograms . . . . .	9-9
9.5.8 Order of DSECTs for histograms . . . . .	9-9
9.5.9 Release level . . . . .	9-10
9.5.10 Number of statistics log records . . . . .	9-10
9.5.11 Output of SREPORT 099 . . . . .	9-10
9.5.12 SREPORT 000 . . . . .	9-10
9.6 Producing statistics reports . . . . .	9-11
9.6.1 Syntax . . . . .	9-11
9.6.2 Parameters . . . . .	9-12
9.6.3 Examples . . . . .	9-13
9.6.4 Operating system considerations . . . . .	9-14
9.7 DC/UCF system statistics reports . . . . .	9-17
9.7.1 SREPORT 003 — IDMS-DC System Statistics . . . . .	9-17
9.7.2 SREPORT 012 — IDMS-DC Task Summary . . . . .	9-27
9.7.3 SREPORT 013 — IDMS-DC Program Summary . . . . .	9-29
9.7.4 SREPORT 014 — IDMS-DC Queue Summary . . . . .	9-31
9.7.5 SREPORT 015 — IDMS-DC Line Summary . . . . .	9-31
9.7.6 SREPORT 016 — IDMS-DC Physical Terminal Summary . . . . .	9-33
9.8 Task and External Request Unit Service (ERUS) statistics reports . . . . .	9-35
9.8.1 Field descriptions . . . . .	9-38
9.9 Transaction statistics reports . . . . .	9-43
9.9.1 Field descriptions . . . . .	9-45
9.10 CA-ADS dialog statistics . . . . .	9-46
9.10.1 Field descriptions . . . . .	9-49
9.11 Histogram Report . . . . .	9-53
9.11.1 Field descriptions . . . . .	9-55
9.12 Record Summary Statistics Report . . . . .	9-57
9.12.1 Contents of SREPORT 017 . . . . .	9-58
9.12.2 Field descriptions . . . . .	9-59
 <b>Chapter 10. Modifying CA-IDMS Reports</b> . . . . .	10-1
10.1 Common modifications . . . . .	10-3

10.2 Steps to modify reports . . . . .	10-4
10.3 Obtaining report module listings . . . . .	10-5
10.3.1 Types of listings . . . . .	10-5
10.4 Coding in CA-CULPRIT . . . . .	10-6
10.4.1 Overview of CA-CULPRIT . . . . .	10-6
10.4.2 CULPRIT coding guidelines . . . . .	10-6
10.4.3 Types of CA-CULPRIT parameters . . . . .	10-6
10.4.3.1 Parameters within each category . . . . .	10-6
10.4.4 Report-specific and global parameters . . . . .	10-8
10.4.5 Edit parameters . . . . .	10-8
10.4.5.1 Functions of each column . . . . .	10-8
10.4.6 Summary of parameter types . . . . .	10-9
10.5 Temporary modifications . . . . .	10-12
10.5.1 Copying the report module . . . . .	10-12
10.5.2 Deleting columns . . . . .	10-12
10.5.3 Changing headings . . . . .	10-13
10.5.4 Selecting specific data . . . . .	10-14
10.5.5 Specifying a new sort sequence . . . . .	10-14
10.5.6 Moving columns . . . . .	10-14
10.6 Permanent modifications . . . . .	10-16
 <b>Chapter 11. Other CA-IDMS Reporting Facilities</b> . . . . .	 11-1
11.1 Overview . . . . .	11-3
11.2 Online reporting facilities . . . . .	11-4
11.2.1 CA-OLQ . . . . .	11-4
11.2.2 IDMS-DC/UCF OnLine PLOG (OLP) . . . . .	11-4
11.2.3 DC/UCF DCMT DISPLAY commands . . . . .	11-5
11.2.4 IDMS-DC/UCF OPER WATCH commands . . . . .	11-5
11.2.5 IDD DDDL DISPLAY commands . . . . .	11-5
11.2.6 CA-IDMS Performance Monitor . . . . .	11-6
11.3 Batch reporting facilities . . . . .	11-7
 <b>Appendix A. OS/390 JCL</b> . . . . .	 A-1
A.1 Running in local mode . . . . .	A-3
A.1.1 Central version modifications . . . . .	A-6
 <b>Appendix B. VSE/ESA JCL</b> . . . . .	 B-1
B.1 Running in local mode . . . . .	B-3
B.1.1 Central version modifications . . . . .	B-4
B.2 IDMSLBLS procedure . . . . .	B-5
B.2.1 What is the IDMSLBLS procedure? . . . . .	B-5
B.3 IDMSLBLS procedure listing . . . . .	B-6
 <b>Appendix C. VM/ESA Commands</b> . . . . .	 C-1
C.1 Running in local mode . . . . .	C-3
C.1.1 Central version modifications . . . . .	C-5
C.1.2 Creating the SYSIPT or SYSIDMS file . . . . .	C-5
 <b>Appendix D. BS2000/OSD JCL</b> . . . . .	 D-1
D.1 Job stream . . . . .	D-3

<b>Appendix E. CA-IDMS Module Listing</b>	E-1
E.1 AREPORT listing	E-4
E.2 CREPORT listing	E-5
E.3 DREPORT listing	E-7
E.4 IREPORT listing	E-10
E.5 JREPORT listing	E-11
E.6 QREPORT listing	E-12
E.7 SREPORT listing	E-13
<b>Index</b>	X-1



## How to Use This Manual

---

## What this manual is about

This manual serves as an introduction to CA-IDMS® reports, which include:

CA-ADS® dialog reports	AREPORTs
CA-IDMS/DC and CA-IDMS/UCF system definition reports	CREPORTs
Dictionary and CA-ICMS™ catalog reports	DREPORTs
ASF row-level security reports	IREPORTs
CA-IDMS/DB journal reports	JREPORTs
CA-IDMS/DB SQL Dictionary Reports	QREPORTs
CA-IDMS/DC and CA-IDMS/UCF (DC/UCF) system statistics reports	SREPORTs

## Who should use this manual

The manual is designed as a reference tool for system, dictionary, and database administrators.

## How the reports are presented

This manual provides a chapter for each type of report. The chapters include:

- A description of and uses for each report category
- Instructions for producing the reports, including examples
- A description of each report, including sample output and field descriptions



## Related documentation

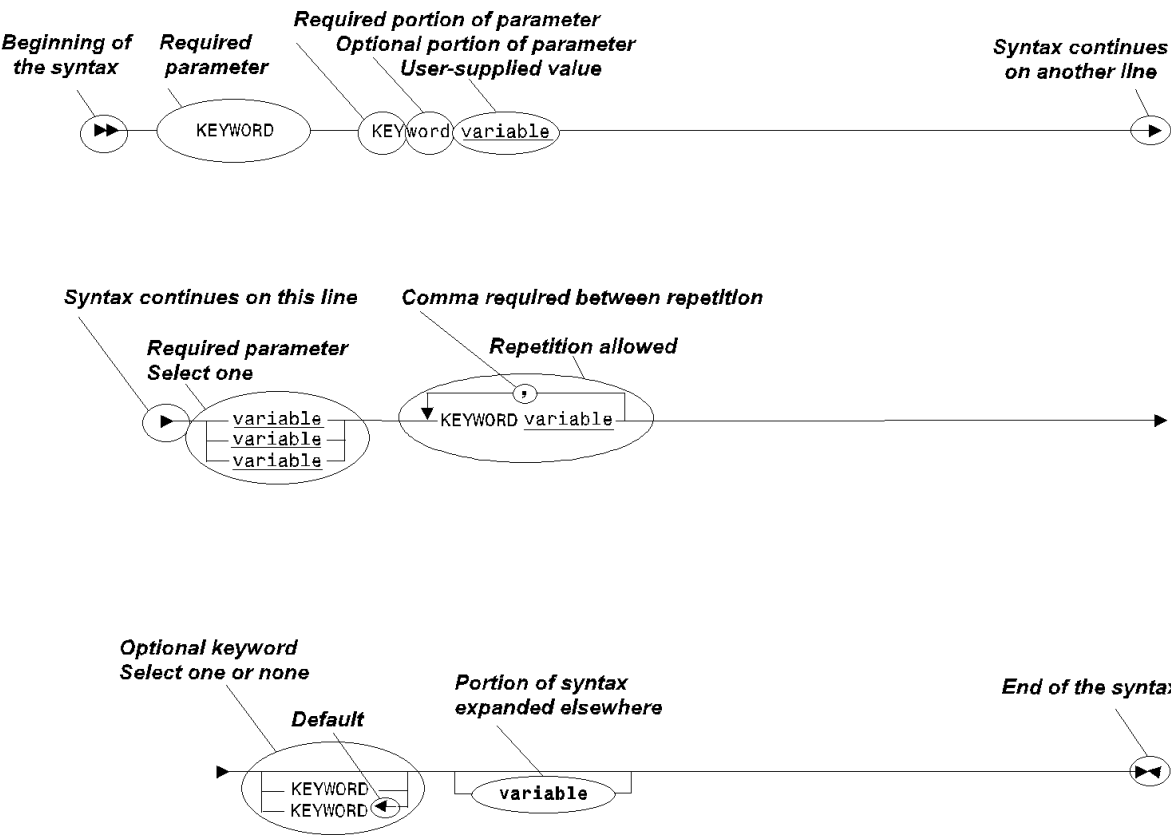
- *CA-ADS Reference*
- *CA-CULPRIT Reference*
- *CA-CULPRIT User Guide*
- *CA-IDMS Utilities*
- *CA-IDMS System Generation*
- *CA-IDMS System Operations*
- *IDD DDDL Reference*
- *&U\$DDR.*
- *Data Dictionary Network Diagram*

# Understanding syntax diagrams

Look at the list of notation conventions below to see how syntax is presented in this manual. The example following the list shows how the conventions are used.

UPPERCASE OR SPECIAL CHARACTERS	Represents a required keyword, partial keyword, character, or symbol that must be entered completely as shown.						
lowercase	Represents an optional keyword or partial keyword that, if used, must be entered completely as shown.						
<u>underlined lowercase</u>	Represents a value that you supply.						
←	Points to the default in a list of choices.						
<b>lowercase bold</b>	Represents a portion of the syntax shown in greater detail at the end of the syntax or elsewhere in the document.						
▶▶	Shows the beginning of a complete piece of syntax.						
◀◀	Shows the end of a complete piece of syntax.						
→	Shows that the syntax continues on the next line.						
▶	Shows that the syntax continues on this line.						
→	Shows that the parameter continues on the next line.						
▶	Shows that a parameter continues on this line.						
▶ parameter →	Shows a required parameter.						
▶ <table><tr><td>parameter</td><td>parameter</td></tr><tr><td>└─┘</td><td>└─┘</td></tr></table> →	parameter	parameter	└─┘	└─┘	Shows a choice of required parameters. You must select one.		
parameter	parameter						
└─┘	└─┘						
▶ <table><tr><td>parameter</td></tr><tr><td>└─┘</td></tr></table> →	parameter	└─┘	Shows an optional parameter.				
parameter							
└─┘							
▶ <table><tr><td>parameter</td><td>parameter</td></tr><tr><td>└─┘</td><td>└─┘</td></tr></table> →	parameter	parameter	└─┘	└─┘	Shows a choice of optional parameters. Select one or none.		
parameter	parameter						
└─┘	└─┘						
▶▶ <table><tr><td>parameter</td></tr><tr><td>└─┘</td></tr></table> →	parameter	└─┘	Shows that you can repeat the parameter or specify more than one parameter.				
parameter							
└─┘							
▶▶ <table><tr><td>parameter</td><td>,</td><td>parameter</td></tr><tr><td>└─┘</td><td></td><td>└─┘</td></tr></table> →	parameter	,	parameter	└─┘		└─┘	Shows that you must enter a comma between repetitions of the parameter.
parameter	,	parameter					
└─┘		└─┘					

# Sample syntax diagram





# Chapter 1. Introduction

---

1.1 Reports stored as dictionary modules . . . . .	1-3
1.2 Dictionary reports . . . . .	1-4
1.3 Run-time reports . . . . .	1-5
1.4 Producing CA-IDMS reports . . . . .	1-6
1.4.1 DATABASE parameter . . . . .	1-6
1.4.2 PARAM= parameter . . . . .	1-7
1.4.3 REPORT= parameter . . . . .	1-9
1.4.4 KEY parameter . . . . .	1-10
1.4.5 SELECT and BYPASS parameters . . . . .	1-10
1.4.5.1 Examples . . . . .	1-13
1.5 Modifying CA-IDMS reports . . . . .	1-15
1.6 CA-CULPRIT Security Considerations . . . . .	1-16



## 1.1 Reports stored as dictionary modules

CA-IDMS reports are stored as modules in the dictionary at installation. The reports document information maintained in the dictionary and statistics collected at system run time. Accordingly, there are two main types of CA-IDMS reports:

- Dictionary reports
- Run-time reports

## 1.2 Dictionary reports

**Uses:** Dictionary reports help dictionary administrators:

- Monitor the contents of dictionaries
- Identify relationships between dictionary entities

**Types of dictionary reports:** There are five types of dictionary reports:

Type	Description	Chapter
DREPORTs	Document information contained in the DDLDDL area of the dictionary, such as basic entities (for example, ELEMENTS ) and teleprocessing entities (for example, LINES). DREPORTs also document information about the CA-ICMS catalog, a directory of information used by the Information Center Management System and the Automatic System Facility (ASF).	Chapter 2, "Standard Dictionary Reports — DREPORTs" and Chapter 3, "CA-ICMS Catalog Reports — DREPORTs"
CREPORTs	Document information in the dictionary associated with DC/UCF systems; for example, physical terminals defined for a particular system.  The DDLDDL, DDLDCMSG, and DDLDCLOD areas of the dictionary supply the information for the reports.	Chapter 4, "DC/UCF System Reports — CREPORTs"
AREPORTs	Document CA-ADS dialogs that are defined to the dictionary and their associated components, such as subschemas, maps, and processes.	Chapter 5, "CA-ADS Reports — AREPORTs"
QREPORTs	Document entities defined to CA-IDMS/DB using SQL commands.	Chapter 6, "CA-IDMS/DB SQL Dictionary Reports — QREPORTs"
IREPORTs	Document ASF table row-level security.	Chapter 7, "ASF Row-Level Security Reports — IREPORTs"



## 1.3 Run-time reports

**Uses:** IDMS-DC/UCF system run-time reports summarize program activity against the database and run-time events. These reports help database administrators (DBAs) and system administrators:

- Monitor database and system performance
- Tune databases and systems
- Research problems, such as broken chains

**Types of run time reports:** There are two types of system run-time reports:

Type	Description	Chapter
JREPORTs	Document the contents of archived journal files; journal files record program activity against the database, such as the before and after images of updated database records.	Chapter 8, "CA-IDMS/DB Journal Reports — JREPORTS"
SREPORTs	Document statistics logged to the DDLCLOG runtime area and off-loaded to the system log file. The statistics keep track of system run-time events, such as the number of times storage was requested within a certain time period.	Chapter 9, "DC/UCF Statistics Reports — SREPORTS"

**Note:** Journal reports are available to all sites where CA-CULPRIT™ and CA-IDMS/DB are installed.

## 1.4 Producing CA-IDMS reports

**Reports execute as batch jobs:** CA-IDMS reports run as batch jobs in either local mode or under the central version. The job control statements required for OS/390, VSE/ESA, VM/ESA, and BS2000/OSD operating systems appear in Appendixes A through D, respectively.

**User-supplied input parameters:** Five user-supplied parameters control report execution:

- DATABASE
- PARAM=
- REPORT=
- KEY
- SELECT/BYPASS

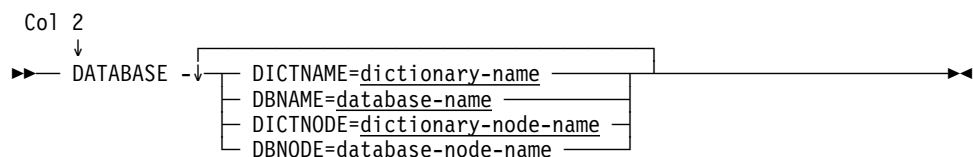
Each of these five parameter types is discussed below, followed by examples. Report-specific considerations are described in the appropriate sections.

### 1.4.1 DATABASE parameter

**Purpose:** DATABASE is an optional parameter that you can use in a multiple dictionary environment or in a multiple system environment:

- In a multiple dictionary environment, you can retrieve report modules from one dictionary and run those modules against another dictionary; for example, obtain the report modules from the system dictionary and report on an application dictionary.
- In a multiple system environment, you can specify the remote nodes that control the dictionaries that contain either the report modules or the data of interest.

#### Syntax



#### Parameters

##### DATABASE

Identifies the parameter. If used, the DATABASE parameter must be the first parameter submitted and must be coded starting in column 2. Only one DATABASE parameter can be specified per run. Any or all of the DATABASE options described below may be coded in any order.

**DICTNAME**

Specifies the name (up to 8 characters) of the dictionary in which the report modules are stored. The following considerations apply:

- The default dictionary is the system dictionary.
- If D-, C-, or AREPORT modules are stored in an application dictionary, the application dictionary must contain the source definitions for IDMSNWKA, the subschema required to run these reports.
- If the report is run under the central version, the specified dictionary must be known to the central version.
- If the report is run in local mode, the specified dictionary must be defined in the database name table load module associated with the DMCL used at runtime.

**DBNAME**

Specifies the name (up to 8 characters) of the dictionary that supplies the input data for D-, C-, and AREPORTs.

**DICTNODE**

Identifies the name of a system defined to the DC/UCF communications network that controls the dictionary that stores the report modules. *Dictionary-node-name* must be the name (up to 8 characters) of a node defined to the DC/UCF system.

**DBNODE**

Specifies the name of a DC/UCF system defined to the communications network that controls the dictionary that supplies the input data for D-, C-, and AREPORTs. *Database-node-name* must be the name (up to 8 characters) of a node defined to the DC/UCF system.

## 1.4.2 PARAM= parameter

**Purpose:** PARAM= is an optional parameter that controls whether report module parameters appear in the two listings associated with report output:

- The Sequential Parameter Listing lists the parameters as they were coded.
- The Input Parameter Listing lists parameter default values and any parameters automatically generated for the report.

By consulting these listings, you can easily code selection criteria for the report and make modifications to the report.

## 1.4 Producing CA-IDMS reports

```
09/08/99                                SEQUENTIAL PARAMETER LISTING                                V15.0 PAGE 1

00 ** SYSIN **                           DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
                                         PARAM=LIST
                                         + INPUT 10000 DB(D) SS=IDMSNWKA,IDMSNTWK,1

INSTALLATION SECURITY OPTION IS YES
CULPRIT/DIRECTORY INTERFACE - IDMS 15.0 CULPRIT 15.0
DICTIONARY SECURITY OPTION IS OFF
AUTO-ATTRIBUTES IS OFF
REPORT REQUEST PARAMETER -               CREPORT=034
                                         + 34$00**** 'C' REPORTS IDMS-DC STANDARD 01/07/99 ROUTINE-CRPT034
                                         + PATHY1 00AK-012 PANEL-118 MAP-098 MAPRCD-125 RCDSYN-079
                                         + 340UTPUT D LP=50
                                         + 345ORT RSYN-NAME-079 RSYN-VER-079 1 MAP-NAME-098 0
                                         + 343 IDMS-DC MAPPING REPORT
                                         + 34410001 'CREPORT 034'
                                         + 34420056 'LISTING OF MAPS BY RECORD NAME'
                                         + 34430056 'RECORD NAME:'
                                         + 34430072 RSYN-NAME-079
                                         + 34440056 'RECORD VERSION:'
                                         + 34440072 RSYN-VER-079 FM 'ZZZZ9'
                                         + 34450001 ' '
                                         + 3451*001 MAP-NAME-098 HH 'MAP NAME'
                                         + 3451*002 MAP-VER-098 FM 'ZZZZ9' HH 'MAP VERSION'
                                         + 3451*004 PANEL-NAME-118 HH 'PANEL NAME'
                                         + 3451*005 PANEL-VER-118 FM 'ZZZZ9' HH 'PANEL VERSION'
                                         + 347010 IF PATH-ID NE 'Y1' DROP
```

Figure 1-1. Sample Sequential Parameter Listing

```
09/08/99                                INPUT PARAMETER LISTING                                V15.0 PAGE 1

*****
INPUT RECORD TYPE BLOCK FILE DESCRIPTION...
*****
INPUT 10000 F 10000 UM(CULLDCLI) SS=IDMSNWKA,IDMSNTWK,0001

*****
PATH START LVL RECORD-NAME ID2 SET-NAME AREA-NAME
*****
Y1 $ 00AK-012 PANEL-118 MAP-098 MAPRCD-125 RCDSYN-079
Y1 001 00039 001 00AK-012 DDLDML
Y1 002 00271 001 PANEL-118 OOAK-PANEL DDLDML
Y1 003 00403 001 MAP-098 PANEL-MAP DDLDML
Y1 004 00627 001 MAPRCD-125 MAP-MAPRCD DDLDML
Y1 005 00711 001 RCDSYN-079 RCDSYN-MAPRCD DDLDML

*****
REC START SIZE TYPE DP FIELD-NAME RECORD-NAME,LEVEL
*****
REC 00403 008 MAP-NAME-098 MAP-098 $$ GENERATED
REC 00411 002 1 MAP-VER-098 MAP-098 $$ GENERATED
REC 00271 032 PANEL-NAME-118 PANEL-118 $$ GENERATED
REC 00303 002 1 PANEL-VER-118 PANEL-118 $$ GENERATED
REC 00001 002 PATH-ID $$ GENERATED
REC 00711 032 RSYN-NAME-079 RCDSYN-079 $$ GENERATED
REC 00743 002 1 RSYN-VER-079 RCDSYN-079 $$ GENERATED
```

Figure 1-2. Sample Input Parameter Listing

### Syntax

Col 2  
↓  
▶▶ PARAM= [ LIST ]  
          [ NOLIST ]  
          [ EJECT ]

### Parameters

**PARAM=**

Must be coded starting in column 2. Any report module parameters that follow PARAM= are either printed or not printed according to the option specified. The parameter can appear more than once in a run.

**LIST**

Prints the parameters on both listings. LIST is the default for SREPORT runs.

**NOLIST**

Does not print the parameters on either listing, unless a parameter contains an error. Parameters in error are printed along with the associated error messages on both listings. NOLIST is the default for D-, C-, A-, I- and JREPORT runs.

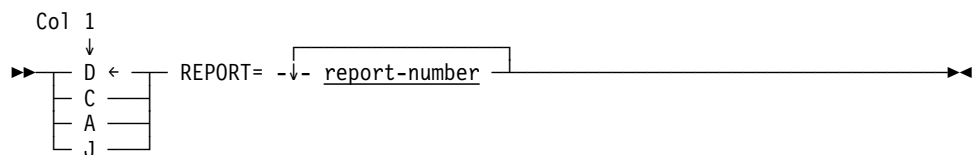
**EJECT**

Prints the parameters at the top of a new page in the Sequential Parameter Listing. EJECT does not affect the pagination of the Input Parameter Listing.

### 1.4.3 REPORT= parameter

**Purpose:** REPORT= is a required parameter for D-, C-, A-, and JREPORTs. The following considerations apply:

- One or more reports of the same type can be specified in a single run.
- When a run specifies one or more D- or CREPORTs, each report must contain PATH parameters that begin with the same database record. A PATH parameter defines a route through the dictionary; The Sequential Parameter Listing above shows a sample PATH parameter.

**Syntax****Parameters****D**

Identifies a dictionary or catalog report. D is the default. If coded, D must be in column 1.

**C**

Identifies a DC/UCF system report. C must be coded in column 1.

**A**

Identifies an CA-ADS dialog report. A must be coded in column 1.

**J**

Identifies a journal report. J must be coded in column 1.

**REPORT=**

Is a keyword that must appear in columns 2 through 8. REPORT= automatically generates an INPUT parameter for A-, C-, and DREPORTs; the INPUT parameter

defines the size of the CA-CULPRIT input buffer and subschema to be accessed. REPORT= also automatically generates REC parameters, which define the fields referenced in the report modules. The Input Parameter Listing above shows sample REC parameters.

**report-number**

Identifies one or more three-digit report-module numbers in the range 000 through 999. Leading zeros can be omitted. Report module numbers must be separated by spaces or commas.

## 1.4.4 KEY parameter

**Purpose:** KEY is a parameter that identifies entity occurrences to be processed. DREPORTs and AREPORTs both include key reports with which you can request detailed information about particular entities. A key report must be the only report specified for the run.

**Syntax**

Col 2  
↓  
► KEY key-field-name -- 'key-field-value' ►

**Parameters****KEY**

(D- and AREPORTS only) Identifies a key report. KEY must be coded starting in column 2. One or more KEY parameters can be coded per report.

**key-field-name**

Specifies the name of the key field. Each key report has a specific key field; Chapter 2, "Standard Dictionary Reports — DREPORTS" and Chapter 5, "CA-ADS Reports — AREPORTS" document the key field names for D- and AREPORTs, respectively.

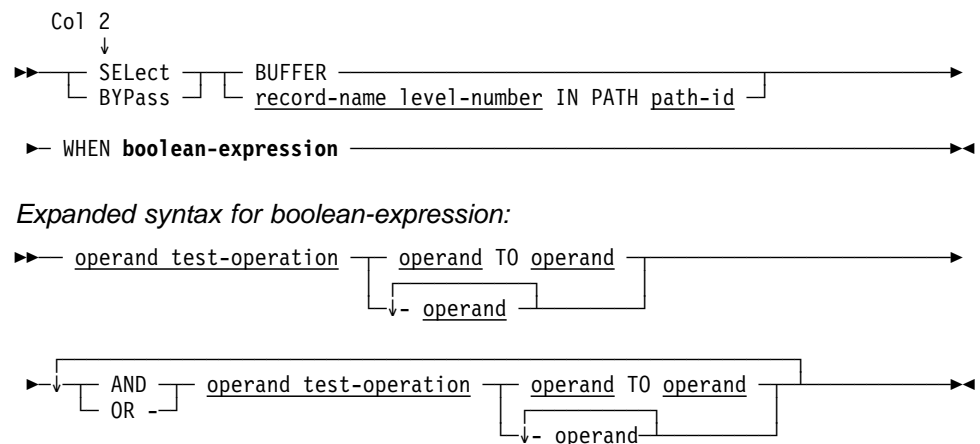
**'key-field-value'**

Specifies one or more values for the key field. *Key-field-value* is an alphanumeric value that must be enclosed in single quotation marks. A list of values must be enclosed in parentheses and values must be separated by spaces or commas.

## 1.4.5 SELECT and BYPASS parameters

**Purpose:** SELECT and BYPASS are optional parameters that select records to be processed based upon specified selection criteria. By using SELECT or BYPASS parameters, you can produce reports that contain only the required information (for example, all programs created after a particular date).

**Syntax**



## Parameters

**SELe**ct

Selects input data. **SELECT** is coded starting in column 2. One or more **SELECT** parameters can be specified per run, but you can not use both **SELECT** and **BYPASS** (see below) in a run.

## BYPass

Bypasses input data. BYPASS is coded starting in column 2. One or more BYPASS parameters can be specified per run, but you can not use both SELECT (see above) and BYPASS in a run.

## BUFFER

Applies the selection criteria to the contents of the completed input buffer (rather than to a specific record type).

**record-name**

(D-, C-, and AREPORTs only) Applies the selection criteria to a specific database record. *Record-name* is the name of a database record, which must also appear on a PATH parameter for the report.

**level-number**

Identifies the occurrence of record in the path; the default is 1.

**IN PATH** path-id

Identifies the path containing the database record. *Path-id* is a 2-byte primary path identifier. If no path id is specified, the selection criteria apply to all preceding PATH parameters that specify the named record.

**WHEN** boolean-expression

Specifies the test criteria to be applied to each record.

**operand**

Specifies an alphanumeric, numeric, or hexadecimal literal or the name of a data item:

- An **alphanumeric literal** is a value (up to 64 characters) that consists of letters, digits, and/or special characters in any combination. An alphanumeric literal must be enclosed in single quotation marks.

- A **numeric literal** is a number (up to 31 digits) that can be preceded by a sign and can contain an embedded or trailing decimal point.
- A **hexadecimal literal** is a hexadecimal string (up to 64 characters) preceded by X and enclosed in single quotation marks (for example, X'0A14').
- A **data-item name** is the name of a field in a record used in report processing. Only names specified in the REC parameter field definitions for the report can be referenced. For sample REC parameters, see the sample Input Parameter Listing earlier in this chapter. For a description of these fields, see the &U\$DDR..

The data type of the left operand must match the data type of the right operand in a boolean expression. To list operands on the right side of the expression:

- Enclose the list in parentheses
- Separate one operand from another with either a blank or a comma

For example:

MAP-NAME-098 EQ ('JKDMP', 'DEHMP', 'TDBMP').

### **test-operation**

Specifies a comparison operator:

- **EQ(E)(=)** -- The value of the left operand is equal to the value of the right operand.
- **NE(N)(#)** -- The value of the left operand is not equal to the value of the right operand.
- **GT(H)(>)** -- The value of the left operand is greater than the value of the right operand.
- **LT(L)(<)** -- The value of the left operand is less than the value of the right operand.
- **GE(>=)(=>)** -- The value of the left operand is greater than or equal to the value of the right operand.
- **LE(<=)(=<)** -- The value of the left operand is less than or equal to the value of the right operand.

### **operand TO operand**

Specifies a range of values to which the left operand is to be compared. *Operand* must be a literal value or the name of a data item, as defined under *operand* above. The specified range must be enclosed in parentheses. For example, MAP-NAME-098 EQ ('A' TO 'G').

### **AND/OR**

Allows the specification of additional test criteria:

- **AND** -- A record is selected for processing if it meets both criteria connected by AND.



- **OR** -- A record is selected for processing if it meets either or both criteria connected by OR.

**Continuing a SELECT or BYPASS parameter:** If the SELECT or BYPASS parameter does not fit on one line, code an asterisk (\*) in column 1 of each continuation line.

### 1.4.5.1 Examples

**Example 1:** This example requests report modules CREPORT 033 and CREPORT 034 from the CULPDICT dictionary. These IDMS-DC/UCF network system reports list maps defined to the system dictionary. The report module parameters are not listed in the output because the user has not requested them.

```
DATABASE DICTNAME=CULPDICT
CREPORT=033,034
```

or

```
CREPORT=33
CREPORT=34
```

Both report modules can execute in the same run because each defines a PATH parameter that starts with the same record type (OOAK-012), as shown in the example below:

```
CREPORT 033 PATHX1 OOAK-012 PANEL-118 MAP-098
CREPORT 034 PATHY1 OOAK-012 PANEL-118 MAP-098 MAPRCD-125 RCDSYN-079
```

**Example 2:** This example is similar to Example 1 except that it selects only those maps defined to the dictionary with a version number of 1 and a name that begins with RQE. (The record and field names for the SELECT parameter can be obtained from either an Input Parameter Listing for CREPORTs 033 and 034 or the &U\$DDR..) Since a path identifier is not specified, the selection criteria apply to both CREPORT 033 and CREPORT 034.

```
DATABASE DICTNAME=CULPDICT
CREPORT=033,034
SELECT MAP-098 WHEN MAP-VER-098 EQ 1 AND
* MAP-NAME-098 EQ ('RQE' TO 'RQF')
```

**Example 3:** In this example, DREPORT 018 reports on map occurrences defined to the DOCUDICT dictionary; the report module itself is stored in the CULPDICT dictionary. The SELECT parameter selects only those CA-CULPRIT input buffers in which users DEH and JKD defined map occurrences. By request, the report outputs the report module parameters.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
PARAM=LIST
DREPORT=18
SELECT BUFFER WHEN PREP-BY-098 EQ ('DEH' 'JKD')
```

**Example 4:** In this example, report module DREPORT 088 is a key report that documents requested map occurrences defined to the DOCUDICT dictionary. The KEY parameter names map occurrences EMPJOB and JKDMAP. The report module parameters are not listed in the report output because the user has not requested them.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
REPORT=88
KEY MAP-NAME-098 ('EMPJOB' 'JKDMAP')
```

**Example 5:** In this example, archive journal records with the program name of IDBCAT are selected for processing. The INPUT parameter defines the physical characteristics of the archive journal file. By request, report module parameters for JREPORT 004 appear in the report output.

```
DATABASE DICTNAME=CULPDICT
INPUT 4276 4276 UM(CULLJRNL)
JREPORT=2
JREPORT=3
PARAM=LIST
JREPORT=4
SELECT PROGRAM-NAME EQ 'IDBCAT'
```

## 1.5 Modifying CA-IDMS reports

CA-IDMS reports are stored as report modules in the dictionary established when CA-IDMS/DB is installed. With some knowledge of CA-CULPRIT, users can modify the existing report modules or create new report modules, as described in Chapter 10, "Modifying CA-IDMS Reports."

Other reporting tools are available to system and database administrators, including both online and batch facilities. Chapter 11, "Other CA-IDMS Reporting Facilities" identifies these reporting tools.

## 1.6 CA-CULPRIT Security Considerations

CA-CULPRIT security is established at several levels. The two main levels to be concerned with to run CA-IDMS reports are installation and product security. Each type of security will be discussed below. For more information on CA-CULPRIT security, see the *CA-CULPRIT Reference*.

**Installation security:** CA-CULPRIT is installed with security either on (enabled) or off. The installation parameter that controls CA-CULPRIT security is CULL-SECURE YES/NO. The default for users who have installed both CA-IDMS and CA-CULPRIT is **No**. CA-CULPRIT must be reinstalled to change the security option. The default for users who have only installed CA-IDMS is **Yes**, which cannot be changed. The Sequential Parameter Listing identifies whether security is in effect with the following statement:

```
C200138 INSTALLATION SECURITY OPTION IS YES (or NO)
```

If security is established at installation time, CA-CULPRIT automatically checks the data dictionary to determine the security level in effect and to enforce that security. If installation security is not established, security options set in the data dictionary will be ignored.

**Product security:** CA-CULPRIT security is established in the data dictionary with the SECURITY FOR CULPRIT IS ON/OFF clause of the SET OPTIONS FOR DICTIONARY statement; the user submits this statement to the DDDL compiler. When security in the dictionary is enabled, CA-CULPRIT validates the user ID and checks all user authorizations. Only authorized users can run CA-CULPRIT jobs that access files or subschemas defined in the data dictionary. A PROFILE parameter must be coded to specify a user ID and password, and must appear before the INPUT parameter.

### Syntax

```
Col 2  
  ↓  
►►— PROFILE USER= user-id PW= password —————►►
```

### Parameters

**USER = user-id**

Specifies the name of a user defined in IDD.

**PW = password**

Specifies the security password associated with a user defined to IDD. The password is not printed in any CA-CULPRIT listings.

## Chapter 2. Standard Dictionary Reports — DREPORTS

---

2.1	Report on the dictionary	2-3
2.2	Information obtained from the DDLDDL area	2-4
2.3	Uses for dictionary reports	2-5
2.4	Dictionary report categories	2-6
2.5	Types of reports	2-7
2.6	Producing dictionary reports	2-8
2.6.1	Syntax	2-8
2.6.2	Parameters	2-8
2.6.3	Key field names for key reports	2-9
2.6.4	Examples	2-10
2.7	Basic entity reports	2-12
2.7.1	System reports (DREPORTs 003, 053, 073)	2-12
2.7.1.1	System Summary Report (DREPORT 053)	2-13
2.7.1.2	System Detail and Key Reports (DREPORTs 003 and 073)	2-13
2.7.2	User reports (DREPORTs 004, 054, 074)	2-15
2.7.2.1	User Summary Report (DREPORT 054)	2-15
2.7.2.2	User Detail and Key Reports (DREPORTs 004 and 074)	2-16
2.7.3	Program reports (DREPORTs 005, 055, 075)	2-20
2.7.3.1	Program Summary Report (DREPORT 055)	2-20
2.7.3.2	Program Detail and Key Reports (DREPORTs 005 and 075)	2-20
2.7.4	Module reports (DREPORTs 006, 056, 076)	2-23
2.7.4.1	Module Summary Report (DREPORT 056)	2-23
2.7.4.2	Module Detail and Key Reports (DREPORTs 006 and 076)	2-24
2.7.5	File reports (DREPORTs 007, 057, 077)	2-26
2.7.5.1	File Summary Report (DREPORT 057)	2-26
2.7.5.2	File Detail and Key Reports (DREPORTs 007 and 077)	2-26
2.7.6	Record reports (DREPORTs 008, 038, 058, 078)	2-29
2.7.6.1	Record Summary Report (DREPORT 058)	2-29
2.7.6.2	Record Detail and Key Report (DREPORTs 008 and 078)	2-29
2.7.6.3	Record/Attribute Key Report (DREPORT 038)	2-32
2.7.7	Element reports (DREPORTs 009, 010, 039, 059, 079)	2-33
2.7.7.1	Element Summary Report (DREPORT 059)	2-34
2.7.7.2	Element Detail and Key Reports (DREPORTs 009 and 079)	2-34
2.7.7.3	Inactive Element Detail Report (DREPORT 010)	2-36
2.7.7.4	Attribute/Element Key Report (DREPORT 039)	2-37
2.8	Site-specific entity reports	2-39
2.8.1	Class Reports (DREPORTs 001 and 071)	2-39
2.8.2	Attribute reports (DREPORTs 002 and 072)	2-41
2.8.3	User-defined entity reports (DREPORTs 019 and 089)	2-43
2.9	Teleprocessing entity reports	2-45
2.9.1	Task reports (DREPORTs 011, 061, 081)	2-46
2.9.1.1	Task Summary Report (DREPORT 061)	2-46
2.9.1.2	Task Detail and Key Reports (DREPORTs 011 and 081)	2-46
2.9.2	Queue reports (DREPORTs 012, 062, 082)	2-48
2.9.2.1	Queue Summary Report (DREPORT 062)	2-48

---

2.9.2.2 Queue Detail and Key Reports (DREPORTs 012 and 082)	2-48
2.9.3 Destination reports (DREPORTs 013, 063, 083)	2-50
2.9.3.1 Destination Summary Report (DREPORT 063)	2-50
2.9.3.2 Destination Detail and Key Reports (DREPORTs 013 and 083)	2-50
2.9.4 Logical terminal reports (DREPORTs 014, 064, 084)	2-52
2.9.4.1 Logical Terminal Summary Report (DREPORT 064)	2-52
2.9.4.2 Logical Terminal Detail and Key Reports (DREPORTs 014 and 084)	2-52
2.9.5 Physical terminal reports (DREPORTs 015, 065, 085)	2-54
2.9.5.1 Physical Terminal Summary Report (DREPORT 065)	2-54
2.9.5.2 Physical Terminal Detail and Key Reports (DREPORTs 015 and 085)	2-55
2.9.6 Line reports (DREPORTs 016, 066, 086)	2-56
2.9.6.1 Line Summary Report (DREPORT 066)	2-56
2.9.6.2 Line Detail and Key Reports (DREPORTs 016 and 086)	2-57
2.9.7 Panel reports (DREPORTs 017, 067, 087)	2-58
2.9.7.1 Panel Summary Report (DREPORT 067)	2-58
2.9.7.2 Panel Detail and Key Reports (DREPORTs 017 and 087)	2-59
2.9.8 Map reports (DREPORTs 018, 068, 088)	2-60
2.9.8.1 Map Summary Report (DREPORT 068)	2-61
2.9.8.2 Map Detail and Key Reports (DREPORTs 018 and 088)	2-61
2.10 Cross-reference reports	2-64
2.10.1 File/Record Cross-Reference Report (DREPORT 020)	2-65
2.10.2 File Synonym Cross-Reference Report (DREPORT 021)	2-66
2.10.3 Record Synonym Cross-Reference Report (DREPORT 022)	2-66
2.10.4 Element Synonym Cross-Reference Report (DREPORT 023)	2-67
2.10.5 Element Description Report (DREPORT 024)	2-68
2.10.6 Element Designator Report (DREPORT 025)	2-70
2.10.7 File Activity Report (DREPORT 026)	2-70
2.10.8 IDMS Set Activity Report (DREPORT 027)	2-71
2.10.9 IDMS Record Activity Report (DREPORT 028)	2-72
2.10.10 IDMS Area Activity Report (DREPORT 029)	2-73
2.10.11 Element/Program Cross-Reference Report (DREPORT 030)	2-74
2.11 Special-purpose report modules	2-76
2.11.1 Comment/Nest Resolution Report Module (DREPORT 00)	2-76
2.11.2 Level Number Report (DREPORT 050)	2-76
2.11.3 Module Text to Card Utility (DREPORT 051)	2-77
2.11.4 Module Text to File Utility (DREPORT 052)	2-78

## 2.1 Report on the dictionary

Standard dictionary reports provide summary, detail, and cross-reference information about the contents of the dictionary, including information about certain entity relationships.

## 2.2 Information obtained from the DDLDDL area

Standard dictionary reports document information maintained in the DDLDDL area of the dictionary. The records in the DDLDDL area of the dictionary are defined and modified by various CA-IDMS system software components, such as the Data Dictionary Definition Language (DDL) compiler and the system generation compiler. Information that is displayed in the fields of the dictionary reports is taken from the corresponding fields in the dictionary records. Note, however, that not all fields in the records are displayed in the standard reports.

►► For more information about how information is defined to the dictionary and about the structure of the records being reported on, see the &U\$DDR..



## 2.3 Uses for dictionary reports

Dictionary reports can:

- Help the MIS department organize and control data by providing an up-to-date source of information about each entity in the dictionary and the relationships between entities
- Help the programmer by providing information such as the names and synonyms of records and their associated elements; task codes and their associated programs; file retention periods; and standardized routines
- Help the DCA manage the communication network by providing information such as the relationship of lines and terminals, the names of programs executed by a task, and the names of users authorized to initiate a task
- Help the systems analyst monitor system performance by providing information on how the existing systems function, whether the systems are being used as originally designed, and the effect a new system could have on an existing system

## 2.4 Dictionary report categories

Dictionary reports are grouped into the following categories for the purpose of discussion:

- **Basic entity reports** document the standard data processing entities: system, user, program, module, file, record, and element.
- **Site-specific entity reports** document entities that are defined to meet the needs of a specific site: class, attribute, and user-defined entities. The system-supplied classes, LANGUAGE and MODE, are included in this category.
- **Teleprocessing entity reports** document entities typical of online systems: destination, line, logical terminal, map, panel, physical terminal, queue, and task.
- **Cross-reference reports** document relationships between specific entities.
- **Special-purpose reports** are used to perform administrative functions, such as punching module source to an output file.

## 2.5 Types of reports

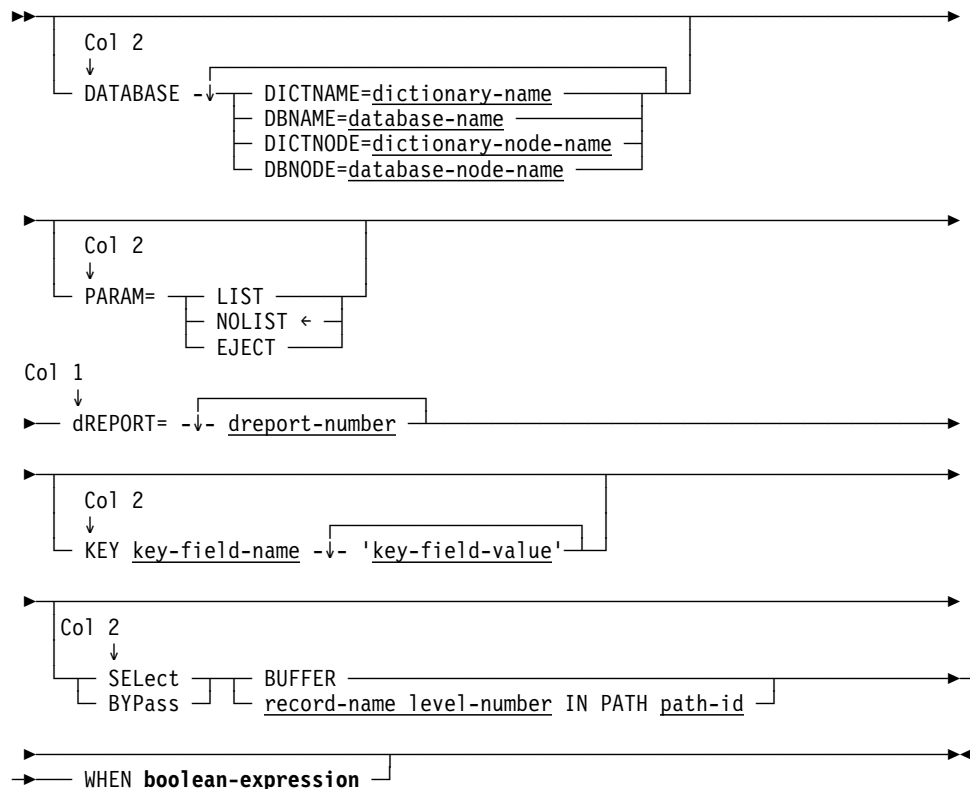
Three types of reports are typically available for each of the basic entity and teleprocessing entity reports:

Type	Number	Description
Summary	053 - 068	List all occurrences of a particular entity type with a minimal amount of supplementary information about each occurrence, such as the version number, description, and the dates the occurrence was defined and updated.
Detail	001 - 019	List every occurrence of a particular entity type with in-depth descriptions of the entity and its relationships with other dictionary entities. You can also run a detail report with a <b>SELECT</b> or <b>BYPASS</b> parameter to report on entities that meet a specified test condition. For example, instead of running a detail report that lists all modules, you could use a <b>SELECT</b> parameter to report on only those entities that were created by a particular user and updated on a particular date.
Key		Are identical to the detail reports in format but list only the entity occurrences specified as key values in the <b>KEY</b> parameter. More than one <b>KEY</b> parameter can be specified for a particular report; each <b>KEY</b> parameter must be specified on a separate line. The syntax for the <b>KEY</b> parameter is described later in this section. The key field names used in the <b>KEY</b> parameter are presented in Table 2-1 on page 2-9.

## 2.6 Producing dictionary reports

A dictionary report is produced by submitting a job that includes the standard Job Control Language (JCL) for CA-CULPRIT report writers and report-specific control statements. JCL for OS/390, VSE/ESA, VM/ESA, and BS2000/OSD operating systems is shown in Appendixes A through D, respectively.

### 2.6.1 Syntax



### 2.6.2 Parameters

General syntax rules for the CA-IDMS reports are presented in Chapter 1, "Introduction." Syntax rules specific to the DREPORTs are described below:

#### **dREPORT dreport-number**

Identifies the report to be run. DREPORT identifies the report as a dictionary report; the D of DREPORT is optional. If D is specified, DREPORT must begin in column 1; if D is not specified, REPORT must begin in column 2.

*Dreport-number* is the 3-digit report number. Leading zeros can be omitted. Dictionary report numbers are shown in Table E-3 on page E-7 and in Tables 2-2 through 2-6 in this section.

Multiple reports can be requested in the same job run with the following exceptions:

- KEY reports cannot be requested in the same run with detail, summary, cross-reference, or special-purpose reports.
- DREPORTs 009 and 010 cannot be run together.

**KEY**

Identifies the key report being requested. KEY must be entered in columns 2-4.

**key-field-name**

Identifies the key field name of the entity type being reported on. Valid key field names for each entity type are listed in Table 2-1.

**'key-value'**

Identifies the entity occurrence of the type specified in *key-field-name*; *key-value* must be enclosed in single quotation marks. If the quoted value is smaller than the field length shown in Table 2-1, CA-CULPRIT pads the value with spaces on the right; if the value is shorter, CA-CULPRIT truncates the value to the specified length.

### 2.6.3 Key field names for key reports

Table 2-1 (Page 1 of 2). DREPORT Key fields

<b>DREPORT Module</b>	<b>DREPORT Name</b>	<b>Name of Key Field</b>	<b>Field Length</b>
038	Attribute/Record Report	ATTR-NAME-093	40
039	Attribute/Element Report	ATTR-NAME-093	40
051	Module Text to Card Utility	MOD-NAME-067	32
052	Module Text to File Utility	MOD-NAME-067	32
071	Class Report	CLASS-NAME-092	20
072	Attribute Report	ATTR-NAME-093	40
073	System Report	SYS-NAME-041	8
074	User Report	USER-NAME-047	32
075	Program Report	PROG-NAME-051	8
076	Module Report	MOD-NAME-067	32
077	File Report	SA-NAM-018	32
078	Record Report	SR-NAM-036	32
079	Element Report	INQ-NAM-058	32
081	Task Report	TASK-NAME-025	8
082	Queue Report	QUEUE-NAME-030	16
083	Destination Report	DEST-NAME-028	8
084	Logical Terminal Report	LTRM-NAME-106	8

Table 2-1 (Page 2 of 2). DREPORT Key fields

DREPORT Module	DREPORT Name	Name of Key Field	Field Length
085	Physical Terminal Report	PTRM-NAME-074	8
086	Line Report	LINE-NAME-109	8
087	Panel Report	PANEL-NAME-118	32
088	Map Report	MAP-NAME-098	8
089	User-Defined Entity Report	CLASS-NAME-092	20

## 2.6.4 Examples

**Example 1:** These control statements can be used to produce a detailed report on the file occurrence ORDER TRANSACTIONS. The report modules used to run the report are found in the default dictionary; data for the report is taken from the DOCUNWK dictionary.

```

DATABASE DBNAME=DOCUNWK
DREPORT=077
KEY SA-NAM-018 'ORDER TRANSACTIONS'
```

**Example 2:** These control statements can be used to produce a file summary report (DREPORT 057) with a parameter listing and a module detail and file detail report without a listing. The report modules used to run the report are in the default dictionary; data for the reports is taken from the DOCUTEST dictionary.

```

DATABASE DBNAME=DOCUTEST
PARAM=LIST
DREPORT=57
PARAM=NOLIST
DREPORT=6,7
```

**Example 3:** These control statements can be used to produce a class report that lists information about the LANGUAGE and DIVISION class occurrences. The key field name for the class report is CLASS-NAME-092; the key values for the report are LANGUAGE and DIVISION. The report modules used to run the report and the data for the report are in the default dictionary.

```

DREPORT=071
KEY CLASS-NAME-092 'LANGUAGE'
KEY CLASS-NAME-092 'DIVISION'
```

**Example 4:** These control statements can be used to produce a module report that lists the source code for dictionary report modules 1, 2, and 3. The report modules used to run the reports and the data for the reports are in the CULPDICT dictionary.

```

DATABASE DICTNAME=CULPDICT DBNAME=CULPDICT
DREPORT=076
KEY MOD-NAME-067 'DREPORT 001'
KEY MOD-NAME-067 'DREPORT 002'
KEY MOD-NAME-067 'DREPORT 003'
```

**Example 5:** These control statements can be used to produce a module detail report that describes only those modules created by user CMH and updated on September 8, 1999. The report modules required to run the report are found in the CULPDICT dictionary; data for the report is taken from the default dictionary.

```
DATABASE DICTNAME=CULPDICT
DREPORT=006
SELECT MOD-NAME-067 WHEN PREP-BY-067 EQ 'CMH' AND
*DATE-LU-067 EQ '09/08/99'
```

**What follows:** The remainder of this section describes the standard dictionary reports. The reports are presented by category and entity type. For example, the MODULE summary, MODULE detail, and MODULE key reports are presented together under 2.7, “Basic entity reports” on page 2-12. Tables 2-2 through 2-6 list the dictionary reports by category; Table E-3 on page E-7 lists the DREPORTs by number.

## 2.7 Basic entity reports

The basic entity reports provide information about seven entity types: system, user, program, module, file, record, and element. Table 2-2 lists the basic entity reports in order of presentation in this section.

---

Table 2-2. Basic Entity Reports. In order of presentation in this section.

---

<b>DREPORT Module</b>	<b>DREPORT Name</b>
003	System Report -- Detail
053	System Report -- Summary
073	System Report -- Key(1)
004	User Report -- Detail
054	User Report -- Summary
074	User Report -- Key(1)
005	Program Report -- Detail
055	Program Report -- Summary
075	Program Report -- Key(1)
006	Module Report -- Detail
056	Module Report -- Summary
076	Module Report -- Key(1)
007	File Report -- Detail(3)
057	File Report -- Summary(3)
077	File Report -- Key(1)
008	Record Report -- Detail
038	Attribute/Record Report -- Key(1)
058	Record Report -- Summary
078	Record Report -- Key(1)
009	Element Report -- Detail(2)
010	Inactive Element Report -- Detail(2)
039	Attribute/Element Report -- Summary
059	Element Report -- Summary
079	Element Report -- Key(1)

(1) KEY reports cannot be run with summary or detail reports.

(2) DREPORTs 009 and 010 cannot be requested in the same run.

(3) DREPORTs 007 and 057 cannot be run with any other reports.

---

### 2.7.1 System reports (DREPORTs 003, 053, 073)

**Purpose:** System reports (DREPORTs 003, 053, and 073) provide information about system occurrences that have been defined to the dictionary. Systems are represented in the dictionary as occurrences of the SYS-041 record.



### 2.7.1.1 System Summary Report (DREPORT 053)

**Contents:** The system summary report provides the following information about systems occurrences:

- System name and version number
- Subsystems associated with the system (indented under the system name)
- Associated description
- Date the system occurrence was added and date updated (if updated)

REPORT NO. 53 DREPORT 053	DATA DICTIONARY REPORTER REL 15.0 SYSTEM REPORT - SUMMARY	09/08/99	PAGE 1
*****			
SYSTEM	DESCRIPTION	----- D A T E -----	
		UPDATED	CREATED
*****			
DCSYSTEM	VER 1		08/11/99
DCSYSTEM	VER 105		07/20/99
DCSYSTEM	VER 9969	07/27/99	03/19/99
LHNSYSTEM	VER 1	08/11/99	08/10/99
LHNTST	VER 1	08/10/99	08/10/99
QATAWDICT	VER 1 QA TAW SYSTEMNAME	08/05/99	08/05/99
TESTSYSTEM	VER 1		08/10/99

Figure 2-1. Sample DREPORT 053

### 2.7.1.2 System Detail and Key Reports (DREPORTs 003 and 073)

**Contents:** The system detail report provides detailed information about all system occurrences; the system key report provides similar information for selected system occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 073. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=073
KEY SYS-NAME-041 'LHNSYSTEM'
```

## 2.7 Basic entity reports

REPORT NO. 73 DREPORT 073	DATA DICTIONARY REPORTER SYSTEM REPORT	REL 15.0	09/08/99	PAGE 1
*****				
SYSTEM NAME	DESCRIPTION	---- D A T E ----		
		UPDATED	CREATED	
*****				
LHNSYSTEM	VER 1	08/11/99	08/10/99	
PREPARED BY	LHN			
REVISED BY	LHN			
PUBLIC ACCESS ALLOWED FOR DISPLAY				
DIVISION	DOCUMENT			
DESTINATION	TESTDEST	VER	1	
LINE	VTAM234	VER	1	
LOGICAL TERMINAL	JESRDR	VER	1	
PHYSICAL TERMINAL	TESTERM	VER	1	
MAP	TEST-MAP	VER	1	
QUEUE	TEST	VER	1	
QUEUE	TESTQUEUE	VER	1	
TASK	TESTBYE	VER	1	
USER LHN				
USER JFD	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION			
USER WITH ACCESS TO THE SYSTEM	LHN			
PROGRAM	EMPSS01	VER	1	
PROGRAM	RHDCBYE	VER	1	
ASSOCIATED TEST SYSTEM	TESTSYSTEM	VER	1	
COMMENT				
00000100	THIS SYSTEM IS USED FOR TEST PURPOSES. IT WILL BE DELETED 09/10/99.			
DEFINITION				
00000100	DELETION DATE			

Figure 2-2. Sample DREPORT 073

**Field descriptions:** The format of DREPORTs 003 and 073 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

### SYSTEM NAME

Identifies the name and version number of the system occurrence being described.

### DESCRIPTION

Identifies text associated with the system occurrence.

### DATE UPDATED/CREATED

Identifies the date the system occurrence was last updated and the date the system occurrence was defined to the dictionary.

### PREPARED BY/REVISED BY

Identifies the user who defined the system occurrence to the dictionary and the user who last updated it.

### PUBLIC ACCESS

Identifies the level of access allowed for unregistered users.

### Class/attribute associations

Identifies class/attribute associations that have been defined for the system occurrence. In this sample report, the attribute DOCUMENT within the class DIVISION is associated with LHNSYSTEM.

### Associated entities

Lists associated entities for the system occurrence with any associated text. The following entity types can be associated with a system occurrence: destination,

line, logical terminal, map, module, physical terminal, process, program, q-file, queue, table, or task.

**USER**

Identifies a user who is registered to access the entity and any responsibility code and text associated with the user. If no users are registered for the entity, the USER literal does not appear; if no responsibility code is associated with the user, the RESPONSIBLE FOR literal does not appear.

**SAME AS**

Identifies the system from which this system definition was copied (if copied).

**User-defined nests**

Identifies user-defined nests that relate one system occurrence to another. In this sample report, ASSOCIATED TEST SYSTEM is a relational key that relates LHNSYSTEM to TESTSYSTEM.

**COMMENT**

Identifies comment text associated with the system occurrence through the COMMENTS clause of the DDDL SYSTEM statement.

**User-defined comments**

Identifies user-defined comments defined through the DEFINITION/*comment-key* clause of the DDDL SYSTEM statement. In this sample report, DELETION DATE is a comment key.

## 2.7.2 User reports (DREPORTs 004, 054, 074)

**Purpose:** User reports (DREPORTs 004, 054, and 074) provide information about user occurrences that have been defined to the dictionary. Users are represented in the dictionary as occurrences of the USER-047 record.

### 2.7.2.1 User Summary Report (DREPORT 054)

**Contents:** The user summary report includes the following information about user occurrences:

- User name
- Associated description
- Date defined and date updated (if updated)

## 2.7 Basic entity reports

REPORT NO. 54 DREPORT 054	DATA DICTIONARY REPORTER REL 15.0 USER REPORT - SUMMARY	09/08/99	PAGE 1
*****			
USER NAME	DESCRIPTION	---- D A T E ---- UPDATED    CREATED	
*****			
BIARI01		07/06/99	
CORP		06/04/99 02/11/99	
CULL DBA		04/07/99 01/08/99	
ENK		06/04/99 05/03/99	
EQA		04/07/99 03/19/99	
FQA		06/04/99 05/27/99	
HANEL01		06/04/99	
IQA		06/22/99	
JFD		08/10/99	
JLK		06/04/99 08/10/99	
LHN	SYSTEM ADMINISTRATOR	08/11/99 08/10/99	
MJH		06/04/99 05/14/99	
MQA		08/09/99	
PAGT001		06/04/99 06/04/99	
PROFUSEA		03/25/99	
PROFUSER		03/19/99	
QAE		04/07/99 02/12/99	
SQA		07/19/99	
UQA		07/27/99	
VQA		08/05/99 08/04/99	
WMF		06/04/99 05/07/99	

Figure 2-3. Sample DREPORT 054

### 2.7.2.2 User Detail and Key Reports (DREPORTs 004 and 074)

**Contents:** The user detail report provides detailed information about all user occurrences; the user key report provides similar information for selected user occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 074. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=074  
KEY USER-NAME-047 'LHN'
```

REPORT NO. 74 DREPORT 074	DATA DICTIONARY REPORTER REL 15.0 USER REPORT	09/08/99      PAGE 1
------------------------------	--------------------------------------------------	----------------------

\*\*\*\*\*  
 USER NAME  
 \*\*\*\*\*

---- D A T E ----  
 UPDATED    CREATED  
 \*\*\*\*\*

LHN		08/11/99 08/10/99
-----	--	-------------------

PREPARED BY	LHN	
REVISED BY	LHN	
DESCRIPTION	SYSTEM ADMINISTRATOR	
FULL NAME	LAURIE NELSON	

AUTHORITIES :		
PASSWORD	NO AUTHORITY	IDD
CULPRIT	NO AUTHORITY	ELEMENT
OLQ	NO AUTHORITY	FILE
ADS	NO AUTHORITY	MODULE
LOAD MODULE	NO AUTHORITY	PROCESS
IDMS		QFILE
SCHEMA	NO AUTHORITY	TABLE
SUBSCHEMA	NO AUTHORITY	PROGRAM
DMCL	NO AUTHORITY	ENTRY POINT
CLASS & ATTRIBUTE		RECORD
CLASS	NO AUTHORITY	REPORT
ATTRIBUTE	NO AUTHORITY	TRANSACTION
		SYSTEM
		USER

		DC
		DESTINATION
		LINE
		LTERM
		MAP
		MESSAGE
		PANEL
		PTERM
		QUEUE
		TASK

DEFAULT OPTIONS	
NO OLQ HEADER	
INTERRUPT	
WHOLE	
FULL	
NO FILLER	
ECHO	
HEADER	
ALL	
COMMENTS	
NO PATH STATUS	
NO CODE TABLE	
NO EXTERNAL PIC	
TERSE	

&&1	
QFILE ALLOWED	
SORT ALLOWED	
OVERRIDES ALLOWED	
MRR ALLOWED	
IDD SIGNON ALLOWED	
SUPERVISOR	JFD
DIVISION	DOCUMENT
DATE-OF-HIRE	08-09-99
PRODUCT	ASF
SYSTEM	LHNSYSTEM      VER    1
	AUTHORIZED TO    ALL AUTHORITY

Figure 2-4. Sample DREPORT 074

## 2.7 Basic entity reports

REPORT NO. 74 DREPORT 074	DATA DICTIONARY REPORTER REL 15.0 USER REPORT	09/08/99	PAGE 2
*****			
USER NAME		----	D A T E ----
		UPDATED	CREATED
*****			
SYSTEM	LHNTST	VER	1
	AUTHORIZED TO	ALL AUTHORITY	
PROGRAM	EMPSS01	VER	1
	AUTHORIZED TO	UPDATE	
ACCESS TO SYSTEM	LHNSYSTEM	VER	1
ACCESS TO SUBSCHEMA	EMPSS01		
OF SCHEMA	EMPSCHM	VER	1
RECORD	EMPOSITION	VER	1
	RESPONSIBLE FOR	CREATION AND UPDATE AND DELETION	
RECORD	EMPOSITION	VER	1
	AUTHORIZED TO	UPDATE	
ELEMENT	LHN-ELEM	VER	1
	RESPONSIBLE FOR	CREATION AND UPDATE AND DELETION	
	AUTHORIZED TO	ALL AUTHORITY	
ELEMENT	DEPT-NAME	VER	1
	RESPONSIBLE FOR	CREATION AND UPDATE AND DELETION	
	AUTHORIZED TO	UPDATE	
DESTINATION	TESTDEST	VER	1
	RESPONSIBLE FOR	CREATION AND UPDATE AND DELETION	
	AUTHORIZED TO	ALL AUTHORITY	
LINE	VTAM234	VER	1
	RESPONSIBLE FOR	CREATION AND UPDATE AND DELETION	
COMMENT			
00000100	USER WAS CREATED FOR TEST PURPOSES		
DEFINITION			
00000100	DELETION DATE		

Figure 2-5. Sample DREPORT 074 - continued

**Field descriptions:** The format of DREPORTs 004 and 074 depends on the order in which information was defined to the dictionary. A description of the fields in the sample user report follows:

### USER NAME

Identifies the name of the user occurrence.

### DATE UPDATED/CREATED

Identifies the date the user occurrence was last updated and the date the user occurrence was defined to the dictionary.

### PREPARED BY/REVISED BY

Identifies the user who defined the user occurrence to the dictionary and the user who last updated it.

### DESCRIPTION

Identifies text associated with the user occurrence.

### FULL NAME

Identifies a name associated with the user occurrence (typically the user's full name).

### PASSWORD ASSIGNED

Indicates that a password is assigned to the user.

### AUTHORITIES

Identifies the user's authority to access secured entities, perform secured operations, or assign or revoke equal authority for another user.

**DEFAULT OPTIONS**

Identifies values that CA-OLQ® uses as default options when the user signs on.

**CA-OLQ options**

Identifies additional CA-OLQ authorizations.

**OVERRIDES ALLOWED**

Indicates whether the user can define and update CA-CULPRIT record layouts and files definitions.

**IDD SIGNON ALLOWED**

Indicates whether the user can access the DDDL compiler.

**IDB ALLOWED**

Indicates that the user can access IDB.

**ASF ALLOWED**

Indicates that the user can access ASF.

**WITHIN USER**

Identifies a relationship between this user occurrence and another user occurrence.

**SAME AS**

Identifies the user occurrence from which this user occurrence was copied (if copied).

**User-defined nests**

Identify user-defined nests that relate one user occurrence to another. In this sample report, the user-defined nest SUPERVISOR relates user LHN to user JFD.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the user occurrence. In this sample report, the attribute ASF within class PRODUCT and the attribute 08/09/99 within class DATE-OF-HIRE are associated with user LHN.

**Associated entity occurrences**

Identifies associated entity occurrences for the user occurrence. Responsibility codes and text associated with each relationship are also listed.

**ACCESS TO**

Indicates that the user is authorized to access a CA-CULPRIT file, a qfile, a signon qfile, a subschema, or a system.

**COMMENT**

Identifies comments associated with the user occurrence through the COMMENTS clause of the DDDL USER statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL USER statement are listed in this field. In this sample report, DELETION DATE is a comment key.

### 2.7.3 Program reports (DREPORTs 005, 055, 075)

**Purpose:** Program reports (DREPORTs 005, 055, and 075) provide information about program occurrences that have been defined to the dictionary. Programs are represented in the dictionary as occurrences of the PROG-051 record.

#### 2.7.3.1 Program Summary Report (DREPORT 055)

**Contents:** The program summary report includes the following information about program occurrences:

- Program name and version number
- Subordinate programs (if any)
- Associated description
- Date defined, date updated (if updated), and date compiled (if compiled). Program occurrences that were created or updated by the DC/UCF system generation compiler do not have these dates associated with them.
- Language associated with the program

REPORT NO. 55			DATA DICTIONARY REPORTER REL 15.0				09/08/99		PAGE 6	
DREPORT 055			PROGRAM REPORT - SUMMARY							
*****										
			----- D A T E -----							
PROGRAM		DESCRIPTION		COMPILED		UPDATED		CREATED		LANGUAGE
*****										
EMPSSAD		VER 1						03/08/99		
EMPSSLR		VER 1						03/08/99		
EMPSSLR1		VER 1						03/25/99		
EMPSS01		VER 1				08/11/99		02/08/99		ASSEMBLER
EMPSS01A		VER 1						03/05/99		
EMPSS02		VER 1						03/08/99		
ERPT01M		VER 1				03/01/99		03/01/99		
GOOD01D		VER 1 ADS DIALOG		07/01/99		07/01/99		07/01/99		
GTRL01D		VER 1 ADS DIALOG		06/11/99		06/11/99		06/11/99		
GWGAPP01		VER 1								
GWGDIA01		VER 1 ADS DIALOG		06/11/99		06/11/99		06/11/99		
GWGDIA02		VER 1 ADS DIALOG		06/30/99		06/30/99		06/30/99		
GWGDIA03		VER 1 ADS DIALOG		03/23/99		03/23/99		03/23/99		
GWGMAP01		VER 1						03/02/99		03/02/99
HELP908		VER 1						00/00/00		05/25/99
H981924		VER 1				00/00/00		06/09/99		
IDMS		VER 1						03/23/99		
IDMSCOBI		VER 1						03/18/99		
IDMSDCCI		VER 1						06/25/99		

Figure 2-6. Sample DREPORT 055

#### 2.7.3.2 Program Detail and Key Reports (DREPORTs 005 and 075)

**Contents:** The program detail report provides detailed information about all program occurrences; the program key report provides similar information about selected program occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 075. The DREPORT and KEY parameters used to create this sample report are:



DREPORT=075  
KEY PROG-NAME-051 'EMPSS01'

REPORT NO. 75 DREPORT 75	DATA DICTIONARY REPORTER REL 15.0 PROGRAM REPORT	09/08/99	PAGE 1
*****			
PROGRAM	ESTIMATED LINES	TIMES COMPILED COMPILE	DATE UPDATED CREATED
*****			
EMPSS01 VER 1	2000	0	08/11/99 02/08/99
PREPARED BY	DPD		
REVISED BY	LHN		
PUBLIC ACCESS ALLOWED FOR ALL	AUTHORITY		
LANGUAGE	ASSEMBLER		
DIVISION	DOCUMENT		
SYSTEM	LHNSYSTEM	VER 1	
USER	LHN		
ASSOCIATED PROGRAM	EMPSS02	VER 1	
COMMENT			
00000100 THIS IS A TEST PROGRAM OF THE EMPLOYEE DATABASE			
DEFINITION			
00000100 DELETION DATE			
MAP USED	TEST-MAP	VER 1	

Figure 2-7. Sample DREPORT 075

**Field descriptions:** The format of DREPORTs 005 and 075 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

#### PROGRAM

Identifies the program name and version number.

#### ESTIMATED LINES

Identifies the estimated number of source code lines in the program.

#### TIMES COMPILED

Identifies the number of times that the program has been compiled. The DML processors automatically update the compile-time statistics if the activity log is on and the dictionary is in UPDATE mode at run time.

#### DATE COMPILED/UPDATED/CREATED

Identifies the date the program occurrence was last compiled or updated and the date the program occurrence was defined to the dictionary.

#### PREPARED BY/REVISED BY

Identifies the user who defined the program to the dictionary and the user who last updated it.

#### PUBLIC ACCESS

Identifies the level of access allowed for unregistered users.

#### DESCRIPTION

Identifies text associated with the program occurrence.

#### Class/attribute associations

Identifies class/attribute associations that have been defined for the program occurrence. In this sample report, the attribute ASSEMBLER within the class

LANGUAGE and the attribute DOCUMENT within class DIVISION are associated with EMPSS01.

**SAME AS**

Identifies the program occurrence from which this program occurrence was copied (if copied).

**SYSTEM**

Identifies a system associated with the program and any user-supplied text for the program/system relationship.

**USER**

Identifies a user who is registered to access the program occurrence and any responsibility codes and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes have been assigned, the RESPONSIBLE FOR literal does not appear.

**SUBPROGRAM**

Identifies a subprogram called by the program.

**CALLED BY**

Identifies a program that calls this program as a subprogram.

**ENTRY**

Identifies an entry point for the program.

**User-defined nests**

Identifies user-defined nests that relate this program occurrence and another program occurrence. In this sample report, the relational key ASSOCIATED PROGRAM relates EMPSS02 to EMPSS01.

**MODULE**

Identifies a module used by the program and any user-supplied text for this program/module relationship.

**RECORD COPIED**

Identifies the records and record synonyms that are copied into the program.

**INPUT FILE**

Identifies the file opened by the program.

**SUBSCHEMA OF SCHEMA**

Identifies a subschema (and associated schema) used by the program.

**Associated logical records, records, sets, and areas**

Specifies associated logical records, records, sets, and areas that the program uses or is expected to use are listed in this field. The field can also show the usage mode for the named area and a count of the number of times the function is performed.

**COMMENT**

Identifies commentary text provided for this program through the COMMENTS clause of the DDDL PROGRAM statement.

**MAP USED**

Identifies the map used by the program.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL PROGRAM statement. In this sample report, DELETION DATE is a comment key.

**ENTRY POINT**

Identifies the program occurrence as an entry point.

**WITHIN PROGRAM**

Identifies the program that uses the program occurrence as an entry point.

## 2.7.4 Module reports (DREPORTs 006, 056, 076)

**Purpose:** Module reports (DREPORTs 006, 056, and 076) provide information about module occurrences that have been defined to the dictionary. Modules, processes, functions, q-files, and edit/code tables are represented in the dictionary as occurrences of the MODULE-067 record.

### 2.7.4.1 Module Summary Report (DREPORT 056)

**Contents:** The module summary report provides the following information about module occurrences:

- Entity name and version number
- Associated description
- Associated language
- Date defined and date updated (if updated)

REPORT NO. 56		DATA DICTIONARY REPORTER REL 15.0		09/08/99		PAGE		8	
DREPORT 056		MODULE REPORT - SUMMARY							
*****									
MODULE NAME		VER DESCRIPTION		LANGUAGE		----- D A T E -----			
						UPDATED		CREATED	
*****									
EMP-COVER-INS-PLAN		1 OLQ		OLQ		06/04/99		02/22/99	
EMP-DEPT-REPORT		1 OLQ		OLQ		08/09/99		08/09/99	
EMPQFILE		1 OLQ		OLQ		07/29/99		07/22/99	
ERPT01P		1		PROCESS				03/01/99	
ERPT01R-AAAA		1		PROCESS				03/01/99	
ERPT02P		1		PROCESS				03/01/99	
ERPT02R-BBBB		1		PROCESS				03/01/99	
ERPT03R-BBBB		1		PROCESS				03/01/99	
FORT77		1		FORTRAN		08/13/99		01/08/99	
GROUPBY-HAVING-COUNT		1 OLQ		OLQ		06/30/99		06/30/99	
GROUPBY-HAVING-ERROR		1 OLQ		OLQ				06/30/99	
GROUPBY-HAVING-JOB		1 OLQ		OLQ		06/30/99		06/30/99	
GROUPBY-WITH-ERROR		1 OLQ		OLQ		06/30/99		06/30/99	
IDMS-DC		1		COBOL		08/13/99		01/08/99	
IDMS-DC-NONAUTO		1		COBOL		08/13/99		08/10/99	
IDMS-SQL-SELECT		1 OLQ		OLQ		07/29/99		07/29/99	
IDMS-STATUS		1		COBOL		08/13/99		01/08/99	
IDMS-STATUS		2		COBOL		08/13/99		01/08/99	
IDMS-STATUS		3		COBOL		08/13/99		01/08/99	
IDMS-STATUS		4		COBOL		08/13/99		01/08/99	
IDMS-WAIT		1		COBOL		08/13/99		01/08/99	
IDMS-WAIT		2		COBOL		08/13/99		01/08/99	

Figure 2-8. Sample DREPORT 056

### 2.7.4.2 Module Detail and Key Reports (DREPORTs 006 and 076)

**Contents:** The module detail report provides detailed information about all module occurrences; the module key report provides similar information about selected module occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 076. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=076  
KEY MOD-NAME-067 'EMPLOYEE'

REPORT NO. 76 DREPORT 076	DATA DICTIONARY REPORTER REL 15.0 MODULE REPORT	09/08/99	PAGE 1
*****			
MODULE NAME		---- D A T E ----	
		UPDATED	CREATED
*****			
EMPLOYEE	VER 1	LANGUAGE OLQ	08/17/99 08/17/99
PREPARED BY	LHN		
REVISED BY	LHN		
DESCRIPTION	TEST MODULE FOR DOCUMENTATION		
PUBLIC ACCESS ALLOWED	FOR ALL AUTHORITY		
USER SQA	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION OLQ		
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
DIVISION	DOCUMENT		
SYSTEM	LHNSYSTEM	VER 1	
SAME AS	EMP-DEPT-REPORT	VER 1	
	LANGUAGE	OLQ	
ASSOCIATED PRODUCTION PROCESS	EMP-DEPT-REPORT	VER 1	
	LANGUAGE	OLQ	
COMMENT			
00000100	THIS IS A TEST MODULE FOR DOCUMENTATION		
DEFINITION			
00000100	DELETION DATE		
MODULE SOURCE			
00000100			
00000200	SET ACCESS OLQ		
00000300	SET DICTNAME ' '		
00000400	SET UNDERLINE '-'		
00000500	SIGNON SS EMPSS01 SCHEMA EMPSCHEM ( 1)		
00000600	OPTIONS ALL HEADER ECHO NOFILLER FULL WHOLE INTERRUPT OLQHEADER -		
00000700	NOPATHSTAT NOSTAT COMMENT VERBOSE NODDBKEY PICTURE CODETAB NOSYN		
00000800	SELECT 'DEPARTMENT','DEPT-ID-0410' 'EMPLOYEE','EMP-ID-0415' 'EMPLOYEE','EMP-FI-		
00000900	RST-NAME-0415' 'EMPLOYEE','EMP-LAST-NAME-0415' 'EMPOSITION','SALARY-AMOUNT-042-		
00001000	0' 'EMPOSITION','BONUS-PERCENT-0420' FROM 'DEPARTMENT', 'EMPLOYEE', 'EMPOSITIO-		
00001100	N' WHERE ('DEPT-EMPLOYEE' AND 'EMP-EMPOSITION')		
00001200	PAGE HEADER BLANK LINES AFTER 1 -		
00001300	LINE 1 'DEPARTMENT/EMPLOYEE/EMPOSITION REPORT' CENTER -		
00001400	LINE 2 '\$DATE' CENTER		
00001500	PAGE FOOTER BLANK LINES BEFORE 1 -		
00001600	LINE 1 '- \$PAGE -' CENTER		
00001700	EDIT DEPT-ID-0410 -		
00001800	ALIGN RIGHT -		
00001900	OLQHEADER 'DEPT'-		
00002000	PICTURE '9999'		
00002100	EDIT EMP-ID-0415 -		
00002200	ALIGN RIGHT -		

Figure 2-9. Sample DREPORT 076

**Field descriptions:** The format of DREPORTs 006 and 076 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

**MODULE NAME**

Identifies the name and version number of the module occurrence.

**DATE UPDATED/CREATED**

Identifies the date the module occurrence was last updated and the date the module occurrence was defined to the dictionary.

**LANGUAGE**

Identifies the language associated with the module occurrence.

**PREPARED BY/REVISED BY**

Identifies the user who defined the module occurrence to the dictionary and the user who last updated it.

**DESCRIPTION**

Identifies the text associated with the module occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

**MODE**

Identifies the operating mode for the module occurrence. MODE is a system-supplied class.

**USER**

Identifies a user who is registered to access this entity and any responsibility code and text associated with the user. If no users are registered for this entity, the USER literal does not appear; if no responsibility code is associated with the user, the RESPONSIBLE FOR literal does not appear.

**SYSTEM**

Identifies the system that includes the module and any user-supplied text associated with the module/system relationship.

**SAME AS**

Identifies the module occurrence from which this module occurrence was copied (if copied).

**PROGRAM**

Identifies the program associated with the module and any user-supplied text for the relationship.

**User-defined nests**

Identify user-defined nests that relate the module occurrence and another module occurrence. In this sample report, the relational key ASSOCIATED PRODUCTION PROCESS is used to relate EMPLOYEE to process EMP-DEPT-REPORT.

**COMMENT**

Identifies commentary text supplied for this module occurrence through the COMMENTS clause of the DDDL MODULE/PROCESS/QFILE/TABLE statement.

**User-defined comments**

Identify user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL

MODULE/PROCESS/QFILE/TABLE statement. In this sample report, DELETION DATE is a comment key.

#### MODULE SOURCE

Lists the source code stored within the module occurrence. The numbers are supplied by the DDDL compiler. If the module occurrence is a q-file, Q-FILE SOURCE appears on the report.

#### CODE TABLE

Identifies the module occurrence as a code table and describes the code table format.

### 2.7.5 File reports (DREPORTs 007, 057, 077)

**Purpose:** File reports (DREPORTs 007, 057, and 077) provide information about IDD files. IDD files are represented in the dictionary as occurrences of the SA-018 record.

#### 2.7.5.1 File Summary Report (DREPORT 057)

**Contents:** The following information is included in the file summary report:

- IDD file name and version number
- Associated text
- Label option
- Date defined and date updated (if updated)

REPORT NO. 57 DREPORT 057	DATA DICTIONARY REPORTER REL 15.0 FILE REPORT - SUMMARY	09/08/99	PAGE 1
*****			
FILE NAME	VER DESCRIPTION	LABELS	---- D A T E ---- UPDATED CREATED
*****			
TESTFILE	1 TEST FILE FOR DOCUMENTATION	08/17/99	08/17/99

Figure 2-10. Sample DREPORT 057

#### 2.7.5.2 File Detail and Key Reports (DREPORTs 007 and 077)

**Contents:** The file detail report provides detailed information about all IDD files; the file key report provides similar information. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 077. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=077
KEY SA-NAM-018 'TESTFILE'
```

REPORT NO. 77 DREPORT 077		DATA DICTIONARY REPORTER REL 15.0 FILE REPORT			09/08/99		PAGE 1	
*****								
FILE NAME	RECORD SIZE	BLOCK SIZE	RECORD FORMAT	LABELS	---- D A T E ---- UPDATED		CREATED	
*****								
TESTFILE	VER 1	132	132	F	NOT SPECIFIED		08/17/99 08/17/99	
PREPARED BY	LHN							
REVISED BY	LHN							
DESCRIPTION	TEST FILE FOR DOCUMENTATION							
PUBLIC ACCESS ALLOWED FOR	ALL AUTHORITY							
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION							
FILE-TYPE	PS							
DEVICE-TYPE	3380							
DIVISION	DOCUMENT							
PROGRAM	LHNPROG	VER 1	I-O FILE					
RECORD	LHN-REC			VER 1				
FILE SYNONYM	TEST-FILE			VER 1				
RECORD SYNONYM	LHN-REC			VER 1				
FILE SYNONYM	LHNFILE			VER 1				
COMMENT								
00000100	THIS IS A DOCUMENTATION TEST FILE							
DEFINITION								
00000100	DELETION DATE							

Figure 2-11. Sample DREPORT 077

**Field descriptions:** The format of DREPORTs 007 and 077 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

#### FILE NAME

Identifies the name and version number of the IDD file occurrence.

#### RECORD SIZE

Identifies the maximum record size for this file occurrence.

#### BLOCK SIZE

Identifies the size (in bytes) of the record blocks on this file.

#### RECORD FORMAT

Identifies whether the record format is fixed (F), undefined (U), variable (V), or variable spanned (S).

#### LABELS

Indicates whether the file labels are NOT SPECIFIED, STANDARD, NON-STANDARD, or OMITTED.

#### DATE UPDATED/CREATED

Identifies the date the IDD file was last updated and the date the file occurrence was defined to the dictionary.

#### PREPARED BY/REVISED BY

Identifies the user who defined the IDD file to the dictionary and the user who last updated it.

#### DESCRIPTION

Identifies text associated with the IDD file occurrence.

**FILE-TYPE**

Identifies the type of IDD file: PS, IS, CARD, VS, or UM.

**VSAM-TYPE**

Identifies the type of VSAM file (if VSAM file): KS, ES, or RS.

**Class/attribute associations**

Identify class/attribute associations that have been defined for the IDD file. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTFILE.

**USER**

Identifies a user who is registered to access the file occurrence and indicates any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are associated with a user, the RESPONSIBLE FOR literal does not appear for that user.

**SAME AS**

Identifies the file occurrence from which this file occurrence was copied (if copied).

**PROGRAM**

Identifies a program that uses this file occurrence for input/output.

**EXTERNAL NAME**

Identifies the ddname or file-id of the file used as input to the program, if defined.

**RELATED FILE**

Identifies a relationship between this file occurrence and another.

**RECORD**

Identifies a record that is associated with this file.

**FILE SYNONYM**

Identifies a file synonym for the file occurrence.

**RECORD SYNONYM**

Identifies record synonym names associated with the last file or file synonym listed.

**COMMENT**

Identifies commentary text supplied for this file occurrence through the COMMENTS clause of the DDDL FILE statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL FILE statement. In this sample report, DELETION DATE is a comment key.

**User-defined nests**

Identifies user-defined nests that relate this file occurrence to another file occurrence.



## 2.7.6 Record reports (DREPORTs 008, 038, 058, 078)

**Purpose:** Record reports (DREPORTs 008, 038, 058, and 078) provide information about record occurrences that have been defined to the dictionary. Elements that are associated with the record are also listed. Record occurrences are represented in the dictionary as occurrences of the SR-036 record.

**Types of record reports:** A summary report, two key reports, and a detail report are provided for record entities. One key report uses the record name as the key value; the other uses an attribute as the key value.

### 2.7.6.1 Record Summary Report (DREPORT 058)

**Contents:** The record summary report provides the following information about record occurrences:

- Record name and version number
- Associated description
- Record length
- Record storage method
- Estimated number of times the record occurs in files or databases
- Date defined and date updated (if updated)

REPORT NO. 58		DATA DICTIONARY REPORTER REL 15.0		09/08/99	PAGE 3
DREPORT 058		RECORD REPORT - SUMMARY			
*****					
RECORD NAME	VER DESCRIPTION	RECORD LGTH STORAGE	OCCURRENCES	---- D A T E ----	UPDATED CREATED
*****					
EMPOSITION	100	28			02/08/99
EXPERTISE	1	8			02/08/99
EXPERTISE	100	8			02/08/99
HOSPITAL-CLAIM	1	292			02/08/99
HOSPITAL-CLAIM	100	292			02/08/99
INSURANCE-PLAN	1	132			02/08/99
INSURANCE-PLAN	100	132			02/08/99
JMAREC	1	4272			03/05/99
JMASQLDA	1	2416			03/29/99
JOB	1	299			02/08/99
JOB	100	299			02/08/99
JOB-IDD	1	4			03/08/99

Figure 2-12. Sample DREPORT 058

### 2.7.6.2 Record Detail and Key Report (DREPORTs 008 and 078)

**Contents:** The record detail report provides detailed information about all record occurrences; the record key report provides similar information about selected record occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 078. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=078  
KEY SR-NAM-036 'LHNREC'

REPORT NO. 78 DREPORT 078		DATA DICTIONARY REPORTER REL 15.0 RECORD REPORT			09/08/99	PAGE 1
*****						
RECORD NAME	RECORD LENGTH	BUILDER	RECORD TYPE	OCCURRENCES	---- D A T E ---- UPDATED	CREATED
*****						
LHNREC	VER 1 28	D			08/19/99	08/19/99
PREPARED BY	LHN					
REVISED BY	LHN					
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY						
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION					
LANGUAGE	COBOL					
LANGUAGE	ASSEMBLER					
LANGUAGE	FORTRAN					
ASSOCIATED RECORD						
TEST-REC			VER 1			
PRIMARY FILE	TESTFILE		VER 1			
DEFINITION						
100 DELETION DATE						
RECORD	LHNREC		VER 1 D			
LANGUAGE	COBOL					
LANGUAGE	ASSEMBLER					
LANGUAGE	FORTRAN					
FILE	TESTFILE		VER 1			
100 02 START-DATE			X(6)		DISPLAY	
VALUE		'YYMMDD'				
200 02 FINISH-DATE					DISPLAY	

Figure 2-13. Sample DREPORT 078

**Field descriptions:** The format of DREPORTs 008 and 078 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

#### RECORD NAME

Identifies the name and version number of the record occurrence.

#### RECORD LENGTH

Identifies the record length in bytes.

#### BUILDER

Identifies the compiler last used to add or update the record occurrence. Possible values are D (DDDL compiler), S (schema compiler), C (mapping compiler), and X (IDMSDIRL).

#### RECORD TYPE

Identifies the record storage mode, if defined.

#### OCCURRENCES

Identifies the actual or estimated number of times the record occurs in files or databases.

#### DATE UPDATED/CREATED

Identifies the date the record occurrence was last updated and the date the record occurrence was added to the dictionary.

**PREPARED/REVISED BY**

Identifies the name of the user who defined the record occurrence and the name of the user who last updated it.

**DESCRIPTION**

Identifies text associated with the record occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

**USER**

Identifies a user who is registered to access the record occurrence and identifies the responsibility code and any associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes have been defined for the user, the RESPONSIBLE FOR literal does not appear.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the record occurrence are listed in this field. In this sample report, the attribute DOCUMENT within class DIVISION is associated with LHNREC.

**SAME AS**

Identifies the record occurrence from which this record occurrence was copied (if copied).

**MODE**

Identifies the operating mode for the program in which the record is used.

**LANGUAGE**

Identifies the language associated with the record occurrence.

**User-defined nests**

Identify user-defined nests that relate the record occurrence to another record occurrence. In this sample report, the relational key ASSOCIATED RECORD relates record LHNREC to record TEST-REC.

**PRIMARY FILE**

Identifies the file associated with the record.

**COMMENT**

Identifies commentary text supplied for the record occurrence through the COMMENTS clause of the DDDL RECORD statement.

**User-defined comments**

Identify user-defined comments that were defined through the DEFINITION/OLQ HEADER/CULPRIT HEADER/*comment-key* clause of the DDDL RECORD statement are listed in this field. In this sample report, DELETION DATE is a comment key.

**RECORD**

Identifies a primary record and provides a description of the elements associated with the record. If defined, the language associated with the record and the programs, files, and subschemas associated with the record are displayed.

**IN SCHEMA/AREA**

Identifies the name and version number of the schema and the area in which the record occurs.

**RECORD SYNONYM**

Identifies a record synonym and provides a description of the elements associated with the record synonym. The associated language is also displayed, if defined.

**SUBSCHEMA**

Identifies the subschema in which this record is used and provides a description of the elements associated with the subschema.

**SUBSCHEMA VIEW**

Indicates that the record is used in a view and provides a description of the elements (fields) used in the view.

### 2.7.6.3 Record/Attribute Key Report (DREPORT 038)

**Contents:** The record/attribute key report provides information about all records associated with a specified attribute. The entries that can appear on this report are identical to those described for the record key report (DREPORT 078) above.

The report below shows sample output for DREPORT 038. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=038  
KEY ATTR-NAME-093 'COBOL'
```

Note that the KEY parameter specifies an *attribute* name rather than a *record* name.

REPORT NO. 38		DATA DICTIONARY REPORTED				REL 15.0		09/08/99		PAGE 1	
DREPORT 038		RECORD REPORT									
*****											
RECORD NAME		RECORD LENGTH		BUILDER		RECORD TYPE		OCCURRENCES		---- D A T E ----	
										UPDATED CREATED	
*****											
KEY ATTRIBUTE: COBOL											
DB-STATISTICS		VER		1		100		C		08/13/99 01/08/99	
PREPARED BY		PUBLIC									
REVISED BY		MJH									
LANGUAGE		COBOL									
RECORD		DB-STATISTICS		VER		1		D			
LANGUAGE		COBOL									
COPIED IN		PROGRAM		LNTN01D		VER		1			
100		03		DATE-TODAY		X(8)				DISPLAY	
200		03		TIME-TODAY		X(8)				DISPLAY	
300		03		PAGES-READ		S9(8)				COMP	
400		03		PAGES-WRITTEN		S9(8)				COMP	
500		03		PAGES-REQUESTED		S9(8)				COMP	
600		03		CALC-TARGET		S9(8)				COMP	
700		03		CALC-OVERFLOW		S9(8)				COMP	
800		03		VIA-TARGET		S9(8)				COMP	
900		03		VIA-OVERFLOW		S9(8)				COMP	
1000		03		LINES-REQUESTED		S9(8)				COMP	
1100		03		RECS-CURRENT		S9(8)				COMP	
1200		03		CALLS-TO-IDMS		S9(8)				COMP	
1300		03		FRAGMENTS-STORED		S9(8)				COMP	
1400		03		RECS-RELOCATED		S9(8)				COMP	
1500		03		LOCKS-REQUESTED		S9(8)				COMP	
1600		03		SEL-LOCKS-HELD		S9(8)				COMP	
1700		03		UPD-LOCKS-HELD		S9(8)				COMP	
1800		03		RUN-UNIT-ID		S9(8)				COMP	
1900		03		TASK-ID		S9(8)				COMP	
2000		03		LOCAL-ID		X(8)				DISPLAY	
2100		03		FILLER		X(8)				DISPLAY	
EMPOSITION		VER		1		28		S		08/11/99 02/08/99	
PREPARED BY		DPD									
REVISED BY		LHN									
USER LHN		RESPONSIBLE FOR CREATION AND UPDATE AND DELETION									
LANGUAGE		COBOL									
ASSOCIATED RECORD											
TEST-REC				VER		1					
IN SCHEMA		EMPSCHM		VER		1		AREA		EMP-DEMO-REGION	
IN SCHEMA		TEST		VER		1		AREA		EMP-DEMO-REGION	
DEFINITION											
100		DELETION DATE									
RECORD		EMPOSITION		VER		1		S		SUFFIX -0420	
LANGUAGE		COBOL									
SUBSCHEMA		EMPSS01		OF SCHEMA		EMPSCHM		VER		1	
SUBSCHEMA		EMPLR01		OF SCHEMA		EMPSCHM		VER		1	
SUBSCHEMA		A202SS01		OF SCHEMA		EMPSCHM		VER		1	
SUBSCHEMA		AD210SS2		OF SCHEMA		EMPSCHM		VER		1	
SUBSCHEMA		AD210SS3		OF SCHEMA		EMPSCHM		VER		1	

Figure 2-14. Sample DREPORT 038

## 2.7.7 Element reports (DREPORTs 009, 010, 039, 059, 079)

**Purpose:** Element reports (DREPORTs 009, 010, 039, 059, and 079) provide information about element occurrences that have been defined to the dictionary. Elements are represented in the dictionary as occurrences of the INQ-058 record. One summary report, two detail reports, and two key reports are provided for the ELEMENT entity type.

### 2.7.7.1 Element Summary Report (DREPORT 059)

**Contents:** The element summary report provides the following information about element occurrences:

- Element name and version number
- The builder code for the element
- Element length
- Element picture
- Element usage mode
- Whether the JUSTIFY option is on
- Whether the BLANK ON ZERO option is in effect
- Whether the SYNC option is on
- The date defined and date updated (if updated)

REPORT NO. 59		DATA DICTIONARY REPORTER REL 15.0					09/08/99		PAGE 16		
DREPORT 059		ELEMENT REPORT - SUMMARY									
*****											
ELEMENT NAME		VERSION	BLDR	LGTH	PICTURE	USAGE	JUST	BONZ	SIGN	----- D A T E ----- UPDATED CREATED	
*****											
EMP-ZIP-FIRST-FIVE	VER	100	D	5	X(5)	DISPLAY				02/08/99 02/05/99	
EMP-ZIP-LAST-FOUR	VER	1	D	4	X(4)	DISPLAY				02/08/99 02/08/99	
EMP-ZIP-LAST-FOUR	VER	100	D	4	X(4)	DISPLAY				02/08/99 02/05/99	
ENTER-HIT	VER	1	D			COND				01/08/99	
ERAREA	VER	1	D	8		COMP-2				01/08/99	
ERAREA	VER	2	D	16	X(16)	DISPLAY				01/08/99	
EREC	VER	1	D	16	X(16)	DISPLAY				01/08/99	
ERRAREA	VER	1	D	16	X(16)	DISPLAY				01/08/99	
ERREC	VER	1	D	8		COMP-2				01/08/99	
ERREC	VER	2	D	16	X(16)	DISPLAY				01/08/99	
ERROR-AREA	VER	1	D	16	X(16)	DISPLAY				01/08/99	

Figure 2-15. Sample DREPORT 059

### 2.7.7.2 Element Detail and Key Reports (DREPORTs 009 and 079)

**Contents:** The element detail report provides detailed information about all element occurrences; the element key report provides similar information about selected element occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 079. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=079
KEY INQ-NAM-058 'DEPT-NAME'
```

REPORT NO. 79 DREPORT 079	DATA DICTIONARY REPORTER REL 15.0 ELEMENT REPORT	09/08/99	PAGE 1
*****			
ELEMENT NAME		BUILD ---- D A T E ----	
*****			
DEPT-NAME	VER 1	D	08/11/99 02/08/99
PREPARED BY	DPD		
REVISED BY	LHN		
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
DIVISION	DOCUMENT		
USER LHN			
DEFINITION	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
100 DELETION DATE			
COMMENT			
100 THIS ELEMENT IS THE DEPARTMENT HEAD IN THE DEPARTMENT RECORD			
ASSOCIATED ELEMENT	DEPT-NAME	VER 100	
PICTURE	X(45)	DISPLAY	LEN= 45
ELEMENT SYNONYM	DEPT-NAME-0410		
RECORD NAME	DEPARTMENT	VER 1	IDD BUILT
RECORD NAME	DEPARTMENT	VER 1	IN SCHEMA EMPSCHM VER 1
RECORD NAME	DEPARTMENT	VER 1	IN SCHEMA TEST VER 1
ELEMENT SYNONYM	DEPTNAME		
RECORD SYNONYM	DEPARTMT	VER 1	
ELEMENT SYNONYM	DPNAME		
RECORD SYNONYM	DEPT	VER 1	
DEPT-NAME	VER 100	D	02/08/99 02/05/99
PREPARED BY	DPD		
REVISED BY	DPD		

Figure 2-16. Sample DREPORT 079

**Field descriptions:** The format of the DREPORTs 009 and 079 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

#### ELEMENT NAME

Identifies the name and version of the element occurrence being described.

#### BUILD CODE

Identifies the compiler that last updated the element occurrence. Possible values are D (DDDL compiler) and S (schema compiler).

#### DATE UPDATED/CREATED

Identifies the date the element occurrence was last updated and the date the element occurrence was added to the dictionary.

#### PREPARED BY/REVISED BY

Identifies the user who defined the element occurrence to the dictionary and the user who last updated it.

#### DESCRIPTION

Identifies text associated with the element occurrence.

#### PUBLIC ACCESS

Identifies the level of access allowed for unregistered users.

### **Class/attribute associations**

Identifies class/attribute associations that have been defined for the element occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with DEPT-NAME.

### **USER**

Identifies a user who is registered to access the element occurrence and any responsibility codes and text associated with the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to a user, the RESPONSIBLE FOR literal does not appear.

### **SAME AS**

Identifies the element occurrence from which this element occurrence was copied (if copied).

### **User-defined nests**

Identifies user-defined nests that relate the element occurrence to another element occurrence. In this sample report, the relational key ASSOCIATED ELEMENT relates DEPT-NAME to version 100 of DEPT-NAME.

### **COMMENT**

Identifies commentary text associated with the element occurrence through the COMMENTS clause of the DDDL ELEMENT statement.

### **User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL ELEMENT statement are listed in this field. In this sample report, DELETION DATE is a comment key.

### **SUBORDINATE ELEMENT**

Identifies a subordinate element of a group element.

### **PRIMARY GROUP**

Identifies the element as a primary group element and identifies the element usage and length.

### **ELEMENT/RECORD NAME**

Identify the element and the record that contains the element or element synonym.

### **WITHIN GROUP**

Identifies the group element that includes this subordinate element.

### **PICTURE/RANGE/VALUE**

Identify the characteristics of the element.

## **2.7.7.3 Inactive Element Detail Report (DREPORT 010)**

**Contents:** The inactive element detail report lists all element occurrences that are not associated with a record occurrence. The entries that can appear on this report are identical to those described for the active element key report (DREPORT 079) above, with the exception of the fields that describe the element/record associations.



REPORT NO. 10 DREPORT 010	DATA DICTIONARY REPORTER ELEMENT REPORT	REL 15.0	09/08/99	PAGE 1
*****				
ELEMENT NAME			BUILD ---- D A T E ----	
*****				
ADXTBIF-MESSAGE	VER 1		D	02/25/99
PREPARED BY	RSB			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY				
PICTURE	X(64)	DISPLAY	LEN= 64	
DATE-ELEM	VER 1		D	02/01/99
PREPARED BY	GCH			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY				
PICTURE	9(8)	DISPLAY	LEN= 8	
DC-ATTN-INT	VER 1		D	01/08/99
PREPARED BY	PUBLIC			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY				
VALUE '4525'				
VALUE '4625'				
WITHIN GROUP	ERROR-STATUS	VER 1		PRIMARY GROUP
PRIMARY GROUP		COND		
ERROR-STATUS	VER 1		D	01/08/99
PREPARED BY	PUBLIC			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY				
VALUE '1400'				
SUBORDINATE ELEMENT	DB-STATUS-OK	VER 1		PRIMARY GROUP

Figure 2-17. Sample DREPORT 010

### 2.7.7.4 Attribute/Element Key Report (DREPORT 039)

**Contents:** The attribute/element key report provides information about all elements that are associated with a specified attribute, regardless of the class association. The entries that can appear on this report are identical to those described for the element key report (DREPORT 079) above.

The report below shows sample output for DREPORT 039. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=039
KEY ATTR-NAME-093 'DOCUMENT'
```

Note that these KEY parameters specify *attribute* names rather than *element* names.

## 2.7 Basic entity reports

REPORT NO. 39 DREPORT 039	DATA DICTIONARY REPORTER REL 15.0 ELEMENT REPORT	09/08/99	PAGE 1
*****			
ELEMENT NAME		BUILD ---- D A T E ----	
*****		CODE UPDATED CREATED	
*****			
KEY ATTRIBUTE: DOCUMENT			
DEPT-NAME	VER 1	D	08/11/99 02/08/99
PREPARED BY	DPD		
REVISED BY	LHN		
DIVISION	DOCUMENT		
USER LHN			
DEFINITION			
	100 DELETION DATE		
COMMENT			
	100 THIS ELEMENT IS THE DEPARTMENT HEAD IN THE DEPARTMENT RECORD		
ASSOCIATED ELEMENT	DEPT-NAME	VER 100	
PICTURE	X(45)	DISPLAY LEN= 45	
ELEMENT SYNONYM	DEPT-NAME-0410		
RECORD NAME	DEPARTMENT	VER 1	IDD BUILT
RECORD NAME	DEPARTMENT	VER 1	IN SCHEMA EMPSCHM VER 1
RECORD NAME	DEPARTMENT	VER 1	IN SCHEMA TEST VER 1
ELEMENT SYNONYM	DEPTNAME		
RECORD SYNONYM	DEPARTMT	VER 1	
ELEMENT SYNONYM	DPNAME		
RECORD SYNONYM	DEPT	VER 1	

Figure 2-18. Sample DREPORT 039

## 2.8 Site-specific entity reports

**Purpose:** The site-specific entity reports provide information about three entity types: class, attribute, and user-defined entities. These entities are typically used to classify entities and to establish relationships between entities beyond the standard relationships provided through IDD.

**Classes supplied at installation:** Two classes, LANGUAGE and MODE, are supplied at installation. Other classes can be defined to further describe your particular environment.

►► For more information about classes, attributes, and user-defined entities, see the *IDD User Guide*.

**Site-specific reports:** Table 2-3 lists the site-specific reports in order of presentation in this section.

Table 2-3. Site-Specific Entity Reports

DREPORT Module	DREPORT Name
001	Class Report -- Detail
071	Class Report -- Key(1)
002	Attribute Report -- Detail
072	Attribute Report -- Key(1)
019	User-Defined Entity Report -- Detail
089	User-Defined Entity Report -- Key(1)

(1) Key reports cannot be run with summary or detail reports.

### 2.8.1 Class Reports (DREPORTs 001 and 071)

**Contents:** Class reports provide information about class occurrences and their associated attributes. Classes are represented in the dictionary as occurrences of the CLASS-092 record; attributes are represented as occurrences of the ATTRIBUTE-093 record. There is no summary report for the class entity type; the detail report is concise and can be used in its place.

The class detail report provides information for all class occurrences defined to the dictionary; the class key report provides similar information for selected class occurrences. The fields and format of these two reports are the same.

## 2.8 Site-specific entity reports

REPORT NO. 01 DREPORT 001	DATA DICTIONARY REPORTER CLASS REPORT	REL 15.0	09/08/99	PAGE 2
*****				
CLASS/ATTRIBUTE	----- D A T E -----	ATTRI	DELETION	CLASS
	UPDATED	CREATED	LOCK	TYPE
*****				
DIVISION	08/11/99	08/10/99	OFF	
PREPARED BY	LHN			
REVISED BY	LHN			
COMMENT				
00000100 THIS IS A TEST CLASS.				
DEFINITION				
00000100 DELETION DATE				
DEVELOPMENT				
DOCUMENT				
ELEMENT DESIGNATOR	08/13/99	01/08/99	A	OFF
PREPARED BY	PUBLIC			
REVISED BY	MJH			
ENTITY	08/13/99	01/08/99	ON	
PREPARED BY	PUBLIC			
REVISED BY	MJH			
ATTRIBUTE				
CLASS				
DESTINATION				
ELEMENT				
FILE				
LINE				
LOGICAL-TERMINAL				
MAP				
MESSAGE				
MODULE				
PANEL				
PHYSICAL-TERMINAL				
PROGRAM				
QUEUE				
RECORD				
SYSTEM				
TASK				
USER				
FILE-TYPE	08/13/99	01/08/99	S	ON
PREPARED BY	PUBLIC			

Figure 2-19. Sample DREPORT 001

**Field descriptions:** The format of DREPORTs 001 and 071 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

**CLASS/ATTRIBUTE**

Identifies the class and/or attribute being described.

**DATE UPDATED/CREATED**

Identifies the date the class occurrence was last updated and the date the class occurrence was defined to the dictionary.

**ATTRI**

Indicates whether attributes are defined automatically (A) or manually (blank) and whether attributes are singular (S) or plural (blank).

**DELETION LOCK**

Indicates whether the deletion lock for this class occurrence is ON or OFF.

**CLASS TYPE**

Indicates whether this class is a CLASS (default) or a user-defined ENTITY.

**PREPARED BY/REVISED BY**

Identifies the user who defined the class occurrence to the dictionary and the user who last updated it.

**COMMENT**

Identifies commentary text associated with the class occurrence through the COMMENTS clause of the DDDL CLASS statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL CLASS statement. In this sample report, DELETION DATE is a comment key.

**Attributes**

Identifies attributes associated with the class occurrence.

## 2.8.2 Attribute reports (DREPORTs 002 and 072)

**Contents:** Attribute reports provide information about attributes defined to the dictionary. Attributes are represented in the dictionary as occurrences of the ATTRIBUTE-093 record. There is no summary report for the attribute entity type; the class detail report may be used in its place.

The attribute detail report provides detailed information about all attribute occurrences; the attribute key report provides similar information about selected attribute occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 072. The DREPORT and KEY parameters used to create this report are:

```
DREPORT=072  
KEY ATTR-NAME-093 'DIVISION'
```

## 2.8 Site-specific entity reports

REPORT NO. 72 DREPORT 072	DATA DICTIONARY REPORTER REL 15.0 ATTRIBUTE REPORT	09/08/99	PAGE 1
*****			
CLASS/ATTRIBUTE	---- D A T E ----	ATTRI	DELETION
	UPDATED CREATED	A S	LOCK
*****			
DIVISION			
DOCUMENT	08/10/99		OFF
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER	LHN		
SYSTEM	LHNSYSTEM	VER	1
PROGRAM	EMPSS01	VER	1
ELEMENT	LHN-ELEM	VER	1
ELEMENT	DEPT-NAME	VER	1
LINE	VTAM234	VER	1
PHYSICAL TERMINAL	TESTERM	VER	1
LOGICAL TERMINAL	JESRDR	VER	1
DESTINATION	TESTDEST	VER	1
QUEUE	TEST	VER	1
QUEUE	TESTQUEUE	VER	1
TASK	TESTBYE	VER	1
PANEL	TEST-PANEL	VER	1
MAP	TEST-MAP	VER	1

Figure 2-20. Sample DREPORT 072

**Field descriptions:** The format of DREPORTs 002 and 072 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

### CLASS/ATTRIBUTE

Identifies the class and attribute being described.

### DATE UPDATED/CREATED

Identifies the date this attribute occurrence was last updated and the date the attribute occurrence was defined to the dictionary.

### ATTRI A/S

Indicates whether attributes in this class are defined automatically (A) or manually (blank) and whether attributes in this class are singular (S) or plural (blank).

### DELETION LOCK

Indicates whether the deletion lock for the attribute is ON or OFF.

### PREPARED/REVISED BY

Identifies the user who defined the attribute occurrence and the user who last updated it.

### PUBLIC ACCESS

Identifies the level of access allowed for unregistered users.

### Associated entity occurrences

Identifies associated entity occurrences that have been defined for the attribute occurrence.

### COMMENT

Identifies commentary text associated with the attribute occurrence through the COMMENTS clause of the DDDL ATTRIBUTE statement.

**User-defined comments**

Identifies user-defined comments that were defined through the  
 DEFINITION/*comment-key* clause of the DDDL ATTRIBUTE statement.

**2.8.3 User-defined entity reports (DREPORTs 019 and 089)**

**Contents:** User-defined entity reports provide information about user-defined entity occurrences that have been defined to the dictionary. User-defined entities are represented in the dictionary as occurrences of the CLASS-092 record. The user-defined entity detail report provides detailed information about all user-defined entity occurrences; the user-defined entity key report provides similar information about selected user-defined entity occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 089. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=089  
 KEY CLASS-NAME-092 'DIVISION'

REPORT NO. 89	DATA DICTIONARY REPORTER	REL 15.0	09/08/99	PAGE 1
DREPORT 089	DIVISION	REPORT		
*****				
DIVISION	----	D A T E	----	DELETION
	UPDATED	CREATED		LOCK
*****				
DEVELOPMENT		08/11/99		OFF
PREPARED BY	LHN			
DOCUMENT		08/19/99 08/10/99		OFF
REVISED BY	LHN			
USER	LHN			
SYSTEM	LHNSYSTEM	VER	1	
MODULE	EMPLOYEE	VER	1	
		LANGUAGE	OLQ	
PROGRAM	EMPSS01	VER	1	
PROGRAM	LHNSS01	VER	1	
FILE	TEST-FILE	VER	1	
QUEUE	TESTQUEUE	VER	1	
TASK	TESTBYE	VER	1	
PANEL	TEST-PANEL	VER	1	
MAP	TEST-MAP	VER	1	
PRODUCT	ASF			
PRODUCT	OLQ			
ASSOCIATED ATTRIBUTE				
DOCTYPE	SYSDOC			
DEFINITION				
00000100	DELETION DATE			

Figure 2-21. Sample DREPORT 089

**Field descriptions:** The format of DREPORTs 098 and 089 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

**Class name**

Identifies the class name of the user-defined entity being described.

**DATE UPDATED/CREATED**

Identifies the date the user-defined entity occurrence was last updated and the date the user-defined entity occurrence was defined to the dictionary.

**DELETION LOCK**

Indicates whether the deletion lock is ON or OFF.

**Attribute name**

Identifies the attribute associated with this user-defined entity.

**PREPARED BY/REVISED BY**

Identifies the user who defined the user-defined entity occurrence to the dictionary and the user who last updated it.

**Associated entities**

Identifies associated entities for the user-defined entity.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the attribute occurrence. In this sample report, the attributes ASF and OLQ within class PRODUCT are associated with attribute DOCUMENT.

**User-defined nests**

Identifies user-defined nests that relate one user-defined entity occurrence to another. In this sample report, the relational key ASSOCIATED ATTRIBUTE relates attribute SYSDOC within class DOCTYPE to DOCUMENT.

**COMMENT**

Identifies commentary text associated with the user-defined entity occurrence through the COMMENTS clause of the DDDL user-defined entity statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL user-defined entity statement. In this sample report, DELETION DATE is a comment key.



## 2.9 Teleprocessing entity reports

**Purpose:** The teleprocessing entity reports provide information about the following entity types: task, queue, destination, logical terminal, physical terminal, line, panel, and map. These entity types correspond to the standard components for online systems.

**Summary of teleprocessing entity reports:** The table below lists the teleprocessing entity reports in the order of presentation in this section.

<b>DREPORT Module</b>	<b>DREPORT Name</b>
011	Task Report -- Detail
061	Task Report -- Summary
081	Task Report -- Key(1)
012	Queue Report -- Detail
062	Queue Report -- Summary
082	Queue Report -- Key(1)
013	Destination Report -- Detail
063	Destination Report -- Summary
083	Destination Report -- Key(1)
014	Logical Terminal Report -- Detail
064	Logical Terminal Report -- Summary
084	Logical Terminal Report -- Key(1)
015	Physical Terminal Report -- Detail
065	Physical Terminal Report -- Summary
085	Physical Terminal Report -- Key(1)
016	Line Report -- Detail
066	Line Report -- Summary
086	Line Report -- Key(1)
017	Panel Report -- Detail
067	Panel Report -- Summary
087	Panel Report -- Key(1)
018	Map Report -- Detail
068	Map Report -- Summary
088	Map Report -- Key(1)

(1) Key reports cannot be run with summary or detail reports.

## 2.9.1 Task reports (DREPORTs 011, 061, 081)

**Purpose:** Task reports (DREPORTs 011, 061, and 081) provide information about the task occurrences defined to the dictionary. Tasks are represented in the dictionary as occurrences of the TASK-025 record.

### 2.9.1.1 Task Summary Report (DREPORT 061)

**Contents:** The task summary report provides the following information about task occurrences:

- Task name and version number
- Associated text
- Date defined and date updated (if updated)

REPORT NO. 61			DATA DICTIONARY REPORTER REL 15.0		09/08/99	PAGE 1
DREPORT 061			TASK REPORT SUMMARY			
*****						
TASK			---- D A T E ----			
			UPDATED		CREATED	
*****						
BYE	VER	1			04/15/99	01/19/99
CLIST	VER	1			04/15/99	01/19/99
CLOD	VER	1			04/15/99	01/19/99
COBINPUT	VER	1				01/19/99
COBTEST	VER	1				01/19/99
DCMT	VER	1			04/15/99	01/19/99

Figure 2-22. Sample DREPORT 061

### 2.9.1.2 Task Detail and Key Reports (DREPORTs 011 and 081)

**Contents:** The task detail report provides detailed information about all task occurrences; the task key report provides similar information about selected task occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 081. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=081
KEY TASK-NAME-025 'TESTBYE'
```

REPORT NO. 81 DREPORT 081	DATA DICTIONARY REPORTER REL 15.0 TASK REPORT	09/08/99	PAGE 1
*****			
TASK		---- D A T E ----	
*****		UPDATED	CREATED
*****		*****	
TESTBYE VER 1		08/11/99	08/11/99
PREPARED BY LHN			
REVISED BY LHN			
DESCRIPTION TEST TASK			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
WITHIN SYSTEM LHNSYSTEM	VER 1		
INVOKES PROGRAM RHDCBYE VER 1			
DIVISION	DOCUMENT		
COMMENT			
00000100	THIS IS A TEST TASK OCCURRENCE		

Figure 2-23. Sample DREPORT 081

**Field descriptions:** The format of DREPORTs 011 and 081 depends on the order in which information was defined to the dictionary. A description of the fields on the sample report follows:

**TASK**

Identifies the name and version number of the task occurrence being described.

**DATE UPDATED/CREATED**

Identifies the date the task occurrence was last updated (if updated) and the date the task occurrence was defined to the dictionary. Task occurrences that are defined through the DC/UCF system generation compiler do not have these dates associated with them.

**PREPARED BY/REVISED BY**

Identifies the user who defined the task occurrence to the dictionary and the user who last updated it.

**DESCRIPTION**

Identifies text associated with the task occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

**USER**

Identifies a user who is registered to access the task occurrence and identifies any responsibility codes and associated text defined for the user. If no users are registered for the task, the USER literal does not appear; if no responsibility codes are associated with a user, the RESPONSIBLE FOR literal does not appear.

**WITHIN SYSTEM...INVOKES PROGRAM**

Identifies a system associated with the task occurrence and the initial program invoked by the task.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the task. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTBYE.

**COMMENT**

Identifies commentary text associated with the task through the COMMENTS clause of the DDDL TASK statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL TASK statement.

## 2.9.2 Queue reports (DREPORTs 012, 062, 082)

**Purpose:** Queue reports (DREPORTs 012, 062, and 082) provide information about queue occurrences that have been defined to the dictionary. Queues are represented in the dictionary as occurrences of the QUEUE-030 record.

### 2.9.2.1 Queue Summary Report (DREPORT 062)

**Contents:** The queue summary report provides the following information about queue occurrences:

- Queue name and version number
- Associated description
- Date defined and date updated (if updated)

REPORT NO. 62			DATA DICTIONARY REPORTER REL 15.0		09/08/99	PAGE 1
DREPORT 062			QUEUE REPORT SUMMARY			
*****						
QUEUE					---- D A T E ----	
					UPDATED    CREATED	
*****						
TEST	VER	1	TEST QUEUE FOR DOCUMENTATION			08/11/99
TESTQUEUE	VER	1	TEST QUEUE FOR DOCUMENTATION			08/11/99

Figure 2-24. Sample DREPORT 062

### 2.9.2.2 Queue Detail and Key Reports (DREPORTs 012 and 082)

**Contents:** The queue detail report provides detailed information about all queue occurrences; the queue key report provides similar information about selected queue occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 082. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=82
KEY QUEUE-NAME-030 'TESTQUEUE'
```

REPORT NO. 82	DATA DICTIONARY REPORTER REL 15.0	09/08/99	PAGE 1
DREPORT 082	QUEUE REPORT		
*****		----- D A T E -----	
QUEUE		UPDATED	CREATED
*****			
TESTQUEUE	VER 1		08/11/99
PREPARED BY LHN			
DESCRIPTION	TEST QUEUE FOR DOCUMENTATION		
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
USER JFD			
WITHIN SYSTEM LHNSYSTEM	VER 1		
DIVISION	DOCUMENT		
COMMENT			
00000100	THIS IS A TEST QUEUE FOR DOCUMENTATION		
DEFINITION			
00000100	DELETION DATE		

Figure 2-25. Sample DREPORT 082

**Field descriptions:** The format of DREPORTs 012 and 082 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

**QUEUE**

Identifies the name and version number of the queue occurrence being described.

**DATE UPDATED/CREATED**

Identifies the date on which the queue occurrence was last updated and the date the queue occurrence was defined to the dictionary. Queue occurrences that were defined through the DC/UCF system generation compiler do not have these dates associated with them.

**PREPARED BY/REVISED BY**

Identifies the user who defined the queue occurrence and the user who last updated it.

**DESCRIPTION**

Identifies text associated with the queue occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

**USER**

Identifies a user who is registered to access the queue occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes have been defined for a user, the RESPONSIBLE FOR literal does not appear.

**WITHIN SYSTEM**

Identifies a system that is associated with the queue occurrence.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the queue occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTQUEUE.

**COMMENT**

Identifies commentary text associated with the queue occurrence through the COMMENTS clause of the DDDL QUEUE statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL QUEUE statement. In this sample report, DELETION DATE is a comment key.

### 2.9.3 Destination reports (DREPORTs 013, 063, 083)

**Purpose:** Destination reports (DREPORTs 013, 063, and 083) provide information about destination occurrences that have been defined to the dictionary. Destinations are represented in the dictionary as occurrences of the DEST-028 record.

#### 2.9.3.1 Destination Summary Report (DREPORT 063)

**Contents:** The destination summary report provides the following information about destination occurrences:

- Destination name and version number
- Associated description
- Date defined and date updated (if updated)

REPORT NO. 63 DREPORT 063	DATA DICTIONARY REPORTER REL 15.0 DESTINATION REPORT SUMMARY	09/08/99	PAGE 1
*****			
DESTINATION		--- D A T E ---	
*****			
		UPDATED	CREATED
*****			
TESTDEST VER 1 TEST DESTINATION		08/11/99	08/11/99

Figure 2-26. Sample DREPORT 063

#### 2.9.3.2 Destination Detail and Key Reports (DREPORTs 013 and 083)

**Contents:** The destination detail report provides detailed information about all destination occurrences; the destination key report provides similar information about selected destination occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 083. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=083
KEY DEST-NAME-028 'TESTDEST'
```

REPORT NO. 83 DREPORT 083	DATA DICTIONARY REPORTER REL 15.0 DESTINATION REPORT	09/08/99	PAGE 1
*****			
DESTINATION		---- D A T E ----	
		UPDATED	CREATED
*****			
TESTDEST VER 1		08/11/99	08/11/99
PREPARED BY LHN			
REVISED BY LHN			
DESCRIPTION TEST DESTINATION			
PUBLIC ACCESS ALLOWED FOR UPDATE			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
WITHIN SYSTEM LHNSYSTEM	VER 1		
DIVISION	DOCUMENT		
COMMENT			
00000100	THIS IS A TEST DESTINATION		

Figure 2-27. Sample DREPORT 083

**Field descriptions:** The format of DREPORTs 013 and 083 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

**DESTINATION**

Identifies the name and version number of the destination being described.

**DATE UPDATED/CREATED**

Identifies the date the destination occurrence was last updated and the date the destination occurrence was defined to the dictionary.

**PREPARED BY/REVISED BY**

Identifies the user who defined the destination occurrence to the dictionary and the user who last updated it.

**DESCRIPTION**

Identifies text associated with this destination occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed to unregistered users.

**USER**

Identifies a user who is registered to access this destination occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to a user, the RESPONSIBLE FOR literal does not appear.

**WITHIN SYSTEM**

Identifies a system associated with this destination occurrence.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for this destination occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTDEST.

**COMMENT**

Identifies commentary text associated with this destination through the COMMENTS clause of the DDDL DESTINATION statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL DESTINATION statement.

**2.9.4 Logical terminal reports (DREPORTs 014, 064, 084)**

**Purpose:** Logical terminal reports (DREPORTs 014, 064, and 084) provide information about logical terminal occurrences that have been defined to the dictionary. Logical terminals are represented in the dictionary as occurrences of the LTRM-106 record.

**2.9.4.1 Logical Terminal Summary Report (DREPORT 064)**

**Contents:** The logical terminal summary report provides the following information about logical terminal occurrences:

- Logical terminal name and version number
- Associated description
- Date defined and date updated (if updated)

REPORT NO. 64 DREPORT 064	DATA DICTIONARY REPORTER REL 15.0 LOGICAL TERMINAL REPORT SUMMARY	09/08/99	PAGE 1
*****			
LOGICAL TERMINAL		--- D A T E ---	
*****		UPDATED	CREATED
*****			
CONSOLE VER 1		04/15/99	01/19/99
JESRDR VER 1	TEST LTERM FOR DOCUMENTATION		08/11/99
UCFLTB1 VER 1			04/15/99
UCFLTB2 VER 1			04/15/99

Figure 2-28. Sample DREPORT 064

**2.9.4.2 Logical Terminal Detail and Key Reports (DREPORTs 014 and 084)**

**Contents:** The logical terminal detail report provides detailed information about logical terminal occurrences; the logical terminal key report provides similar information about selected logical terminal occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 084. The DREPORT and KEY parameter used to create this sample report are:

```
DREPORT=084
KEY LTRM-NAME-106 'JESRDR'
```



REPORT NO. 84 DREPORT 084	DATA DICTIONARY REPORTER REL 15.0 LOGICAL TERMINAL REPORT	09/08/99	PAGE 1
*****			
LOGICAL TERMINAL		---- D A T E ----	
*****			
JESRDR VER 1			08/11/99
PREPARED BY LHN			
DESCRIPTION TEST LTERM FOR DOCUMENTATION			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
WITHIN SYSTEM	LHNSYSTEM	VER 1	
DIVISION	DOCUMENT		
COMMENT			
00000100 THIS IS A TEST LOGICAL TERMINAL			
DEFINITION			
00000100 DELETION DATE			

Figure 2-29. Sample DREPORT 084

**Field descriptions:** The format of DREPORTs 014 and 084 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

#### LOGICAL TERMINAL

Identifies the name and version number of the logical terminal being described.

#### DATE UPDATED/CREATED

Identifies the date the logical terminal occurrence was last updated and the date the logical terminal occurrence was defined to the dictionary.

#### PREPARED BY/REVISED BY

Identifies the user who defined the logical terminal occurrence to the dictionary and the user who last updated it.

#### DESCRIPTION

Identifies text associated with the logical terminal occurrence.

#### PUBLIC ACCESS

Identifies the level of access allowed for unregistered users.

#### USER

Identifies a user who is registered for access to the logical terminal occurrence and any responsibility code and associated text defined for the user. If no users are registered for this occurrence, the USER literal does not appear; if no responsibility codes have been defined for the user, the RESPONSIBLE FOR literal does not appear.

#### WITHIN SYSTEM...PHYSICAL TERMINAL

Identifies a system and the physical terminals associated with the logical terminal occurrence.

#### Class/attribute associations

Identifies class/attribute associations that have been defined for the logical terminal occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with JESRDR.

**COMMENT**

Identifies commentary text associated with the logical terminal through the COMMENTS clause of the DDDL LOGICAL-TERMINAL statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL LOGICAL-TERMINAL statement. In this sample report, DELETION DATE is a comment key.

## 2.9.5 Physical terminal reports (DREPORTs 015, 065, 085)

**Purpose:** Physical terminal reports (DREPORTs 015, 065, and 085) provide information about physical terminal occurrences that have been defined to the dictionary. Physical terminals are represented in the dictionary as occurrences of the PTRM-074 record.

### 2.9.5.1 Physical Terminal Summary Report (DREPORT 065)

**Contents:** The physical terminal summary report provides the following information about physical terminal occurrences:

- Physical terminal name and version number
- Associated text
- Date defined and date updated (if updated)

REPORT NO. 65 DREPORT 065			DATA DICTIONARY REPORTER REL 15.0 PHYSICAL TERMINAL REPORT SUMMARY		09/08/99	PAGE 1
*****						
PHYSICAL TERMINAL					---- D A T E ----	
*****					UPDATED    CREATED	
*****						
JESRDR	VER	1			01/19/99	
OPERATOR	VER	1			04/15/99	01/19/99
UCFPTB1	VER	1			04/15/99	
UCFPTB2	VER	1			04/15/99	

Figure 2-30. Sample DREPORT 065

2.9.5.2 Physical Terminal Detail and Key Reports (DREPORTs 015 and 085)

**Contents:** The physical terminal detail report provides detailed information about physical terminal occurrences; the physical terminal key report provides similar information about selected physical terminal occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 085. The DREPORT and KEY parameters used to create this sample report are shown below:

DREPORT=085  
KEY PTRM-NAME-074 'TESTTERM'

REPORT NO. 85 DREPORT 085	DATA DICTIONARY REPORTER REL 15.0 PHYSICAL TERMINAL REPORT	09/08/99	PAGE 1
*****			
PHYSICAL TERMINAL		----	DATE ----
		UPDATED	CREATED
*****			
TESTERM VER 1		08/11/99	08/11/99
PREPARED BY LHN			
REVISED BY LHN			
DESCRIPTION TEST PHYSICAL TERMINAL FOR DOCUMENTATION			
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
WITHIN SYSTEM LHN SYSTEM	VER 1		
DIVISION DOCUMENT			
COMMENT			
00000100 THIS IS A TEST PHYSICAL TERMINAL			
DEFINITION			
00000100 DELETION DATE			

Figure 2-31. Sample DREPORT 085

**Field descriptions:** The format of DREPORTs 015 and 085 depends on the order in which the information was defined to the dictionary. A description of the fields in the sample report follows:

**PHYSICAL TERMINAL**

Identifies the name and version number of the physical terminal being described.

**DATE UPDATED/CREATED**

Identifies the date the physical terminal occurrence was last updated and the date the physical terminal occurrence was defined to the dictionary. Physical terminal occurrences that are defined through the DC/UCF system generation compiler do not have these dates associated with them.

**PREPARED BY/REVISED BY**

Identifies the user who defined the physical terminal occurrence to the dictionary and the user who last updated it.

**DESCRIPTION**

Identifies text associated with the physical terminal occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

**USER**

Identifies a user who is registered to access the physical terminal occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned, the RESPONSIBLE FOR literal does not appear.

**WITHIN SYSTEM ...LINE**

Identifies a system associated with the physical terminal and any associated lines.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the physical terminal occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTTERM.

**COMMENT**

Identifies commentary text associated with the physical terminal through the COMMENTS clause of the DDDL PHYSICAL-TERMINAL statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL PHYSICAL-TERMINAL statement are listed in this field. In this sample report, DELETION DATE is a comment key.

## **2.9.6 Line reports (DREPORTs 016, 066, 086)**

**Purpose:** Line reports (DREPORTs 016, 066, and 086) provide information about line occurrences that have been defined to the dictionary. Lines are represented in the dictionary as occurrences of the LINE-109 record.

### **2.9.6.1 Line Summary Report (DREPORT 066)**

**Contents:** The line summary report provides the following information about line occurrences:

- Line name and version number
- Associated description
- Date defined and date updated (if updated)

REPORT NO. 66 DREPORT 066			DATA DICTIONARY REPORTER REL 15.0 LINE REPORT SUMMARY		09/08/99	PAGE 1		
*****								
LINE						----	D A T E	----
						UPDATED	CREATED	
*****								
CONSOLE	VER	1					04/15/99	01/19/99
JESRDR	VER	1						01/19/99
UCFLINE	VER	1					04/15/99	01/19/99
VTAMLIN	VER	1					01/21/99	01/19/99

Figure 2-32. Sample DREPORT 066

2.9.6.2 Line Detail and Key Reports (DREPORTs 016 and 086)

**Contents:** The line detail report provides detailed information about all line occurrences defined to the dictionary; the line key report provides similar information about selected line occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 086. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=86  
KEY LINE-NAME-109 'VTAM234'

REPORT NO. 86 DREPORT 086		DATA DICTIONARY REPORTER REL 15.0 LINE REPORT		09/08/99	PAGE 1
*****					
LINE				---- D A T E ---- UPDATED CREATED	
*****					
VTAM234 VER 1				08/11/99	07/29/99
PREPARED BY LHN					
REVISED BY LHN					
DESCRIPTION TEST LINE OCCURRENCE					
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY					
USER LHN		RESPONSIBLE FOR CREATION AND UPDATE AND DELETION			
WITHIN SYSTEM LHNSYSTEM		VER 1			
DIVISION		DOCUMENT			
COMMENT					
00000100 THIS IS A TEST LINE OCCURRENCE					

Figure 2-33. Sample DREPORT 086

**Field descriptions:** The format of DREPORTs 016 and 086 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

**LINE**

Identifies the name and version number of the line being described.

**DATE UPDATED/CREATED**

Identifies the date the line occurrence was last updated and the date the line occurrence was defined to the dictionary.

**PREPARED BY/REVISED BY**

Identifies the user who defined the line occurrence to the dictionary and the user who last updated it.

**DESCRIPTION**

Identifies text associated with the line occurrence.

**PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

**USER**

Identifies a user who is registered to access the line occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned for a user, the RESPONSIBLE FOR literal does not appear.

**WITHIN SYSTEM**

Identifies a system associated with the line occurrence.

**Class/attribute associations**

Identifies class/attribute associations that have been defined for the line occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with VTAM234.

**COMMENT**

Identifies commentary text associated with the line occurrence through the COMMENTS clause of the DDDL LINE statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL LINE statement.

## **2.9.7 Panel reports (DREPORTs 017, 067, 087)**

**Purpose:** Panel reports (DREPORTs 017, 067, and 087) provide information about panel occurrences that have been defined to the dictionary. Panels are represented in the dictionary as occurrences of the PANEL-118 record.

### **2.9.7.1 Panel Summary Report (DREPORT 067)**

**Contents:** The panel summary report provides the following information about panel occurrences:

- Panel name and version number
- Associated description
- Date defined and date updated (if updated)

REPORT NO. 67 DREPORT 067	DATA DICTIONARY REPORTER REL 15.0 PANEL REPORT SUMMARY	09/08/99	PAGE 1
*****			
PANEL		---- D A T E ----	
		UPDATED	CREATED
*****			
ABCD01M-OLMPANEL	VER 1	00/00/00	08/12/99
ABIF01M-OLMPANEL	VER 1	06/30/99	02/26/99
ADDS01M-OLMPANEL	VER 1	05/04/99	05/03/99
ADMI01M-OLMPANEL	VER 1	08/13/99	02/26/99
ADMS01M-OLMPANEL	VER 1	08/13/99	02/26/99
ADOLQLNM-OLMPANEL	VER 1	02/26/99	02/25/99
ADOL01M-OLMPANEL	VER 1	00/00/00	05/14/99

Figure 2-34. Sample DREPORT 067

2.9.7.2 Panel Detail and Key Reports (DREPORTs 017 and 087)

**Contents:** The panel detail report provides detailed information about all panel occurrences; the panel key report provides similar information about selected panel occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 087. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=087  
KEY PANEL-NAME-118 'TEST-PANEL'

REPORT NO. 87 DREPORT 087	DATA DICTIONARY REPORTER REL 15.0 PANEL REPORT	09/08/99	PAGE 1
*****			
PANEL		---- D A T E ----	
		UPDATED	CREATED
*****			
TEST-PANEL	VER 1	08/11/99	08/11/99
PREPARED BY	LHN		
REVISED BY	LHN		
DESCRIPTION	TEST PANEL FOR DOCUMENTATION		
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
MAP	TEST-MAP	VER 1	
WITHIN SYSTEM	LHNSYSTEM	VER 1	
DIVISION	DOCUMENT		
COMMENT			
00000100	THIS IS A TEST PANEL		
DEFINITION			
00000100	DELETION DATE		

Figure 2-35. Sample DREPORT 087

**Field descriptions:** The format of DREPORTs 017 and 087 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

PANEL

Identifies the name and version number of the panel being described. The -OLMPANEL suffix indicates that the panel occurrence was defined to the

dictionary by the OLM compiler. The suffix -AUTOPANEL indicates that a panel was defined to the dictionary through the AUTOPANEL option of the DC/UCF mapping compiler.

### **DATE UPDATED/CREATED**

Identifies the date the panel occurrence was last updated and the date the panel occurrence was defined to the dictionary. Panel occurrences that are defined through the DC/UCF system generation compiler do not have these dates associated with them.

### **PREPARED BY/REVISED BY**

Identifies the user who defined the panel occurrence to the dictionary and the user who last updated it.

### **DESCRIPTION**

Identifies text associated with the panel occurrence.

### **PUBLIC ACCESS**

Identifies the level of access allowed for unregistered users.

### **USER**

Identifies a user who is registered to access the panel occurrence and any responsibility codes and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to the user, the RESPONSIBLE FOR literal does not appear.

### **MAP...WITHIN SYSTEM**

Identifies a map associated with the panel occurrence and the system associated with the map.

### **Class/attribute associations**

Identifies class/attribute associations that have been defined for the panel occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TEST-PANEL.

### **COMMENT**

Identifies commentary text associated with the panel occurrence through the COMMENTS clause of the DDDL PANEL statement.

### **User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL PANEL statement. In this sample report, DELETION DATE is a comment key.

## **2.9.8 Map reports (DREPORTs 018, 068, 088)**

**Purpose:** Map reports (DREPORTs 018, 068, and 088) provide information about map occurrences that have been defined to the dictionary. Maps are represented in the dictionary as occurrences of the MAP-098 record.



2.9.8.1 Map Summary Report (DREPORT 068)

- Contents:** The map summary report provides the following information about map occurrences:
- Map name and version number
  - Associated description
  - Date defined and date updated (if updated)

REPORT NO. 68 DREPORT 068	DATA DICTIONARY REPORTER REL 15.0 MAP REPORT SUMMARY	09/08/99	PAGE 1
*****			
MAP		----	D A T E ----
*****		UPDATED	CREATED
*****			
ABCD01M VER 1		00/00/00	08/12/99
ABIF01M VER 1		06/30/99	02/26/99
ADDS01M VER 1		05/04/99	05/03/99
ADMI01M VER 1		08/13/99	02/26/99
ADMS01M VER 1		08/13/99	02/26/99
ADOLQ1M VER 1		02/26/99	02/25/99
ADOLQ1M VER 1		05/14/99	05/14/99
ADPT01M VER 1		03/01/99	03/01/99
ADRP01M VER 1	DEFAULT RESPONSE PTF 87-06-1041	07/14/99	02/26/99
ADSL01M VER 1		03/01/99	03/01/99
ADXXMBIF VER 1		02/25/99	02/25/99

Figure 2-36. Sample DREPORT 068

2.9.8.2 Map Detail and Key Reports (DREPORTs 018 and 088)

**Contents:** The map detail report provides detailed information about all map occurrences; the map key report provides similar information about selected map occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 088. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=88  
KEY MAP-NAME-098 'TEST-MAP'

REPORT NO. 88 DREPORT 088	DATA DICTIONARY REPORTER REL 15.0 MAP REPORT	09/08/99	PAGE 1
*****			
MAP		---- D A T E ----	
		UPDATED	CREATED
*****			
TEST-MAP VER 1			08/11/99
PREPARED BY LHN			
DESCRIPTION	THIS IS A TEST MAP FOR DOCUMENTATION		
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY			
USER LHN	RESPONSIBLE FOR CREATION AND UPDATE AND DELETION		
PROGRAM	EMPSS01	VER 1	
WITHIN SYSTEM	LHNSYSTEM	VER 1	
DIVISION	DOCUMENT		
COMMENT			
00000100	THIS IS A TEST MAP FOR DOCUMENTATION		
DEFINITION			
00000100	DELETION DATE		

Figure 2-37. Sample DREPORT 088

**Field descriptions:** The format of DREPORTs 018 and 088 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

#### MAP

Identifies the name and version number of the map being described.

#### DATE UPDATED/CREATED

Identifies the date the map occurrence was last updated and the date the map occurrence was defined to the dictionary.

#### PREPARED BY/REVISED BY

Identifies the user who defined the map occurrence to the dictionary and the user who last updated it.

#### DESCRIPTION

Identifies text associated with the map occurrence.

#### PUBLIC ACCESS

Identifies the level of access allowed for unregistered users.

#### USER

Identifies a user who is registered to access the map occurrence and any responsibility codes and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to the user, the RESPONSIBLE FOR literal does not appear.

#### PROGRAM

Identifies a program that uses the map occurrence.

#### WITHIN SYSTEM

Identifies a system associated with the map occurrence.

#### Class/attribute associations

Identify class/attribute associations that have been defined for the map occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TEST-MAP.

**RECORD**

Identifies a record that contains record elements used in the map occurrence.

**COMMENT**

Identifies commentary text associated with the map occurrence through the COMMENTS clause of the DDDL MAP statement.

**User-defined comments**

Identifies user-defined comments that were defined through the DEFINITION/*comment-key* clause of the DDDL MAP statement. In this sample report, DELETION DATE is a comment key.

## 2.10 Cross-reference reports

**Purpose:** The cross-reference reports provide information about:

- Files and areas and their associated records (DREPORT 020)
- Files and their associated synonyms (DREPORT 021)
- Records and their associated synonyms (DREPORT 022)
- Elements and their associated synonyms (DREPORT 023)
- Elements with their associated descriptions (DREPORT 024)
- Element designators/attribute relationships (DREPORT 025)
- Program use of files, sets, records, and areas (DREPORTs 026, 027, 028, 029)
- Program/element relationships (DREPORT 030)

**Summary of cross-reference reports:** The table below lists the cross-reference reports in order of presentation in this section.

DREPORT Number	DREPORT Name
020	File/Record Cross-Reference Report(1)
021	File Synonym Cross-Reference Report(1)
022	Record Synonym Cross-Reference Report
023	Element Synonym Cross-Reference Report
024	Element Description Report
025	Element Designator Report
026	File Activity Report(1)
027	IDMS Set Activity Report
028	IDMS Record Activity Report
029	IDMS Area Activity Report
030	Element/Program Cross-Reference Report
(1) DREPORTs 020, 021, and 026 cannot be run with any other reports.	

## 2.10.1 File/Record Cross-Reference Report (DREPORT 020)

**Contents:** The File/Record Cross-Reference report lists all IDD file/record relationships defined to the dictionary. The file/record relationships are defined through the RECORD SYNONYM FOR FILE SYNONYM clause or the WITHIN FILE clause of the DDDL RECORD statement.

REPORT NO. 20 DREPORT 020		DATA DICTIONARY REPORTER REL 15.0 FILE/RECORD REPORT				09/08/99	PAGE 1
*****							
FILE NAME		RECORD LENGTH	BLOCK SIZE	LABELS	---- D A T E ---- UPDATED CREATED		
*****							
TESTFILE	VER 1	TEST FILE FOR DOCUMENTATION		132	132	08/17/99 08/17/99	
PRIMARY RECORD		RECORD LHN-REC			VER 1		

Figure 2-38. Sample DREPORT 020

**Field descriptions:** A description of the fields in the sample report follows:

### FILE NAME

Identifies the primary name and version number of an IDD file and displays a description if one is defined for this file occurrence.

### RECORD LENGTH

Identifies the maximum record length for the file.

### BLOCK SIZE

Identifies the block size of the file, if defined.

### LABELS

Identifies any labels defined for the file.

### DATE UPDATED/CREATED

Identifies the date the file occurrence was last updated and the date the file occurrence was defined to the dictionary.

### Description

Identifies text associated with the file occurrence.

### PRIMARY RECORD/RECORD

Identifies the names and version numbers of primary records associated with the file.

### FILE NAME

Identifies the name and version number of a file associated with the named primary record and any text associated with this file occurrence.

### Associated records and elements

Identifies records and elements that are associated with an IDD file occurrence.

```

REPORT NO. 21                      DATA DICTIONARY REPORTER  REL 15.0          09/08/99          PAGE 1
DREPORT 021                      FILE SYNONYM CROSS REFERENCE REPORT

*****
FILE SYNONYM NAME                  PRIMARY FILE NAME
*****

TESTFILE                          VER      1          **SAME**
LHNFILE                          VER      1          TESTFILE                          VER      1

```

Identifies the primary file name and version number for this file synonym or displays **\*\*SAME\*\*** if the file synonym name is the primary file name.

### 2.10.3 Record Synonym Cross-Reference Report (DREPORT 022)

REPORT NO. 22			DATA DICTIONARY REPORTER REL 15.0		09/08/99		PAGE 1	
DREPORT 022			RECORD SYNONYM CROSS REFERENCE REPORT					
*****								
RECORD SYNONYM NAME			PRIMARY RECORD NAME			SCHEMA		
*****								
AD21D008-TEST-RECORD			VER	1	**SAME**			
AD68DEMD-MAP-RECORD			VER	1	**SAME**			
A20209M6-MAP-RECORD-1			VER	1	**SAME**			
A20209M6-MAP-RECORD-2			VER	1	**SAME**			
A20209M6-MAP-RECORD-3			VER	1	**SAME**			
BIFX01M-REC01			VER	1	**SAME**			
BIFX02M-REC01			VER	1	**SAME**			
BLANK-LINE			VER	1	**SAME**			
CALCAB			VER	1	**SAME**			
CBDCM30M			VER	1	**SAME**			
CBTCF01R			VER	1	**SAME**			
CCFP01DX-RECN			VER	1	**SAME**			
COOB01DX-RECN			VER	1	**SAME**			
COOP01D-RECN			VER	1	**SAME**			
COOP02D-RECN			VER	1	**SAME**			
COUNTING			VER	1	**SAME**			
COVERAGE			VER	1	**SAME**			
COVERAGE			VER	1	**SAME**			
COVERAGE			VER	1	**SAME**			
COVERAGE			VER	100	**SAME**			
			</					

Figure 2-40. Sample DREPORT 022

**Field descriptions:** A description of the fields in the sample report follows:

**RECORD SYNONYM NAME**

Identifies the record synonym name and and version number.

**PRIMARY RECORD NAME**

Identifies the primary name and version number for this synonym or displays  
\*\*SAME\*\* if the record synonym name is the primary record name.

**SCHEMA**

Identifies the schema that includes the record, if defined.

## 2.10.4 Element Synonym Cross-Reference Report (DREPORT 023)

**Contents:** The Element Synonym Cross-Reference report lists element occurrences defined to the dictionary with their associated element synonym names. Element synonyms are defined through the ELEMENT NAME SYNONYM clause of the DDDL ELEMENT or RECORD ELEMENT statements and through the schema compiler.

REPORT NO. 23	DATA DICTIONARY REPORTER REL 15.0	09/08/99	PAGE 19
DREPORT 023	ELEMENT SYNONYM CROSS REFERENCE REPORT		
*****			
ELEMENT SYNONYM NAME	PRIMARY ELEMENT NAME	GROUP SYNONYM NAME	
*****			
DENTIST-ZIP-LAST-FOUR	**SAME**	VER	1
DENTIST-ZIP-LAST-FOUR	**SAME**	VER	100
DENTIST-ZIP-LAST-FOUR-0405	DENTIST-ZIP-LAST-FOUR	VER	1
DENTIST-ZIP-LAST-FOUR-0405	DENTIST-ZIP-LAST-FOUR	VER	100
DENTIST-ZIP-0405	DENTIST-ZIP	VER	1
DENTIST-ZIP-0405	DENTIST-ZIP	VER	100
DEOS	**SAME**	VER	1
DEP-COST	**SAME**	VER	1
DEP-COST	**SAME**	VER	100
DEP-COST-0435	DEP-COST	VER	1
DEP-COST-0435	DEP-COST	VER	100
DEPENDENT	**SAME**	VER	1
DEPENDENT	**SAME**	VER	100
DEPENDENT-0400	DEPENDENT	VER	1
DEPENDENT-0400	DEPENDENT	VER	100
DEPT-HEAD-ID	**SAME**	VER	1
DEPT-HEAD-ID	**SAME**	VER	100
DEPT-HEAD-ID-MAP	DEPT-HEAD-ID	VER	100
DEPT-HEAD-ID-0410	DEPT-HEAD-ID	VER	1
DEPT-HEAD-ID-0410	DEPT-HEAD-ID	VER	100
DEPT-ID	**SAME**	VER	1
DEPT-ID	**SAME**	VER	100

Figure 2-41. Sample DREPORT 023

**Field descriptions:** A description of the fields in the sample report follows:

**ELEMENT SYNONYM NAME**

Identifies the name and version number of an element synonym.

**PRIMARY ELEMENT NAME**

Identifies the primary element name associated with this synonym name or displays **\*\*SAME\*\*** if the element synonym name is the primary element name.

**GROUP SYNONYM NAME**

Identifies the group synonym name associated with the element synonym, if defined.

## 2.10.5 Element Description Report (DREPORT 024)

**Contents:** The Element Description report lists element descriptions and all elements associated with each description. This report sorts elements by description. Element description text is defined to the dictionary through the **ELEMENT DESCRIPTION IS** clause of the DDDL **ELEMENT** statement.



REPORT NO. 24	DATA DICTIONARY REPORTER REL 15.0				09/08/99	PAGE 1
ELEMENT DESCRIPTION REPORT						
*****						
DESCRIPTION/DATA ELEMENT NAME	VERSION	LGTH	PICTURE	USAGE	ELEMENT SYNONYM NAME	
*****						
CALCULATED BONUS WK-BONUS-AMOUNT	VER	1	9 9(7)V99	DISPLAY	WK-BONUS-AMOUNT	
CALCULATED TIME EMPLOYED WK-EMP-LENGTH	VER	1		GROUP	WK-EMP-LENGTH	
LHN-ELEM	VER	1		GROUP	EMPID	
EMPLOYEE ID WITHIN EMPLOYEE RECORD LHN-ELEM	VER	1		GROUP	LHN-ELEM	
MONTHS EMPLOYED WK-EMP-MONTH	VER	1	2 99	DISPLAY	WK-EMP-MONTH	
MONTHS IN THE JOB WK-POS-MONTH	VER	1	2 99	DISPLAY	WK-POS-MONTH	
START DATE DAY WK-START-DAY	VER	1	2 99	COMP-3	WK-START-DAY	
START DATE MONTH WK-START-MONTH	VER	1	2 99	COMP-3	WK-START-MONTH	
START DATE YEAR WK-START-YEAR	VER	1	2 99	COMP-3	WK-START-YEAR	
SYSTEM DATE DAY WK-SYSTEM-DAY	VER	1	2 99	COMP-3	WK-SYSTEM-DAY	
SYSTEM DATE MONTH WK-SYSTEM-MONTH	VER	1	2 99	COMP-3	WK-SYSTEM-MONTH	
SYSTEM DATE YEAR WK-SYSTEM-YEAR	VER	1	2 99	DISPLAY	WK-SYSTEM-YEAR	

Figure 2-42. Sample DREPORT 024

**Field descriptions:** A description of the fields in the sample report follows:

**DESCRIPTION/DATA ELEMENT NAME**

Identifies an element description and all elements with that description.

**VERSION**

Identifies the version number of the element.

**LGTH**

Identifies the length in bytes of this data element.

**PICTURE**

Identifies the picture clause associated with this data element.

**USAGE**

Identifies the usage mode of the listed element.

**ELEMENT SYNONYM NAME**

Identifies the element synonym name associated with the element.

## 2.10.6 Element Designator Report (DREPORT 025)

**Contents:** The Element Designator report lists all attributes within the element designator class and describes all elements associated with each element designator. Element designator is a system-provided class that allows you to classify similar elements for report purposes. Element/element designator relationships are defined through `INCLUDE class IS attribute` clause of the DDDL ELEMENT statement, where the *class* is ELEMENT DESIGNATOR.

REPORT NO. 25 DREPORT 025	DATA DICTIONARY REPORTER REL 15.0 ELEMENT DESIGNATOR REPORT				09/08/99	PAGE 1
*****						
DESIGNATOR/DATA ELEMENT NAME	VERSION	LGTH	PICTURE	USAGE	ELEMENT SYNONYM NAME	
*****						
DOCUMENTATION LHN-DEPT-NAME	VER	1	45	X(45)	DISPLAY	LHN-DEPT-NAME

Figure 2-43. Sample DREPORT 025

**Field descriptions:** A description of the fields in the sample report follows:

### DESIGNATOR

Identifies the name of an attribute associated with the element designator class.

### DATA ELEMENT NAME

Identifies the names and version numbers of data elements associated with this element designator.

### LGTH

Identifies the length in bytes of the element.

### PICTURE

Identifies the picture clause associated with this data element.

### USAGE

Identifies the usage mode for the element.

### ELEMENT SYNONYM NAME

Identifies a synonym name associated with the element by the ELEMENT NAME SYNONYM clause of the DDDL ELEMENT or RECORD ELEMENT statements, or by the schema compiler.

## 2.10.7 File Activity Report (DREPORT 026)

**Contents:** The File Activity report provides information about how IDD files are used by programs. These file/program relationships are defined through the INPUT/I-O/OUTPUT clause of the DDDL PROGRAM statement and through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

REPORT NO. 26 DREPORT 026	DATA DICTIONARY REPORTER REL 15.0 FILE ACTIVITY REPORT	09/08/99	PAGE 1
*****			
FILE NAME	PROGRAM	USAGE	REFERENCED SYNONYM NAME
*****			
TESTFILE	VER 1 LHNPROG VER 1 I-0	1	LHNFILE

Figure 2-44. Sample DREPORT 026

**Field descriptions:** A description of the fields in the sample report follows:

**FILE NAME**

Identifies the primary name and version number of a file that is opened by a program.

**PROGRAM**

Identifies the name and version number of a program that accesses this file.

**USAGE**

Indicates whether the program opens the named file for input, output, or input/output operations.

**REFERENCED**

Identifies the number of OPEN statements for this file and usage within the named program.

**SYNONYM NAME**

Identifies the name used by the program to reference the file.

## 2.10.8 IDMS Set Activity Report (DREPORT 027)

**Contents:** The IDMS Set Activity report provides information about how programs reference sets. These set/program relationships are defined through the SET clause of the DDDL PROGRAM statement or through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

REPORT NO. 27 DREPORT 027	DATA DICTIONARY REPORTER REL 15.0 IDMS SET ACTIVITY REPORT	09/08/99	PAGE 1							
*****										
SCHEMA	VER	SUBSCHEMA	SET	PROGRAM	VER	USAGE	TIMES	---- D A T E ----	COMPILED	CREATED
*****										
EMPSCHM	1	EMPSS01	DEPT-EMPLOYEE	NRDY99D	1	OBTAIN	1	03/12/99	03/12/99	
EMPSCHM	1	EMPSS01	DEPT-EMPLOYEE	NRDY97D	1	OBTAIN	1	03/12/99	03/12/99	
EMPSCHM	100	EMPSS01	CALC	CBDMLO1	1	OBTAIN	1	06/25/99	03/23/99	
EMPSCHM	100	EMPSS01	CALC	CBDMLO3	1	FIND	1	03/23/99	03/23/99	
EMPSCHM	100	EMPSS01	CALC	CBDMLO3	1	FIND	5	03/23/99	03/23/99	
EMPSCHM	100	EMPSS01	CALC	CBDMLO3	1	OBTAIN	1	03/23/99	03/23/99	
EMPSCHM	100	EMPSS01	CALC	CBDMLO3	1	OBTAIN	4	03/23/99	03/23/99	
EMPSCHM	100	EMPSS01	DEPT-EMPLOYEE	MCMT03D	1	FIND	1	03/02/99	03/02/99	
EMPSCHM	100	EMPSS01	DEPT-EMPLOYEE	MCMT03D	1	IF	4	03/02/99	03/02/99	
EMPSCHM	100	EMPSS01	DEPT-EMPLOYEE	MCMT04D	1	FIND	1	03/02/99	03/02/99	

Figure 2-45. Sample DREPORT 027

**Field descriptions:** A description of the fields in the sample report follows:

**SCHEMA/VER**

Identifies the name and version number of a schema that contains a set referenced by a program.

**SUBSCHEMA**

Identifies the name of the subschema in which the set exists.

**SET**

Identifies the name of a set associated with a program.

**PROGRAM/VER**

Identifies the name and version number of the program using this set.

**USAGE**

Identifies the DML command issued against this set by the named program.

**TIMES**

Indicates the number of times this DML command is issued against the set within the named program.

**DATE COMPILED/CREATED**

Identifies the date the program was last compiled and the date the program occurrence was defined to the dictionary.

## 2.10.9 IDMS Record Activity Report (DREPORT 028)

**Contents:** The IDMS Record Activity report provides information about how programs use records. These record/program relationships are defined through the RECORD clause of the DDDL PROGRAM statement or through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

REPORT NO. 28 DREPORT 028			DATA DICTIONARY REPORTER REL 15.0 IDMS RECORD ACTIVITY REPORT				09/08/99		PAGE 1	
*****										
SCHEMA VER SUBSCHEMA RECORD				PROGRAM VER USAGE		----- D A T E -----				
						TIMES COMPILED CREATED				
*****										
EMPSCHM	1	AD99SLR	COVERAGE	LRDA01D	1	BIND	1	03/23/99	03/23/99	
EMPSCHM	1	AD99SLR	DEPARTMENT	LRDA01D	1	BIND	1	03/23/99	03/23/99	
EMPSCHM	1	AD99SLR	EMPLOYEE	LRDA01D	1	BIND	1	03/23/99	03/23/99	
EMPSCHM	1	AD99SLR	OFFICE	LRDA01D	1	BIND	1	03/23/99	03/23/99	
EMPSCHM	1	EMPSSLR	DEPARTMENT	LRTD01D	1	BIND	1	03/29/99	03/29/99	
EMPSCHM	1	EMPSSLR	EMPLOYEE	LRTD01D	1	BIND	1	03/29/99	03/29/99	
EMPSCHM	1	EMPSSLR	EMPLOYEE	REN001D	1	BIND	1	08/11/99	08/11/99	
EMPSCHM	1	EMPSSLR	EMPLOYEE	REN001D	1	OBTAIN	2	08/11/99	08/11/99	
EMPSCHM	1	EMPSSLR	EMPLOYEE	REN002D	1	BIND	1	08/11/99	08/11/99	

Figure 2-46. Sample DREPORT 028

**Field descriptions:** A description of the fields in the sample report follows:

**SCHEMA/VER**

Identifies the name and version number of a schema that contains a record referenced by a program.

**SUBSCHEMA**

Identifies the subschema in which the record is included.

**RECORD**

Identifies a record associated with a program.

**PROGRAM/VER**

Identifies the name and version number of the program using the record.

**USAGE**

Identifies the DML command issued against the record by the named program.

**TIMES**

Indicates the number of times the DML command is issued against the record within the named program.

**DATE COMPILED/CREATED**

Identifies the date the program was last compiled and the date the program occurrence was defined to the dictionary.

## 2.10.10 IDMS Area Activity Report (DREPORT 029)

**Contents:** The IDMS Area Activity report provides information about how programs use areas. These area/program relationships are defined through the AREA clause of the DDDL PROGRAM statement or through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

REPORT NO. 29 DREPORT 029		DATA DICTIONARY REPORTER REL 15.0 IDMS AREA ACTIVITY REPORT				09/08/99		PAGE 1
*****								
SCHEMA	VER	SUBSCHEMA	AREA	PROGRAM	VER	ACTIVITY	TIMES	---- D A T E ---- COMPILED CREATED
*****								
EMPSCHM	1	AD99SLR	EMP-DEMO-REGION	LRDA01D	1	RETRIEVAL	3	03/23/99 03/23/99
EMPSCHM	1	AD99SLR	INS-DEMO-REGION	LRDA01D	1	RETRIEVAL	3	03/23/99 03/23/99
EMPSCHM	1	AD99SLR	ORG-DEMO-REGION	LRDA01D	1	RETRIEVAL	3	03/23/99 03/23/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	LRTD01D	1	RETRIEVAL	2	03/29/99 03/29/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	LRTD01D	1	UPDATE	1	03/29/99 03/29/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	OTFT03D	1	RETRIEVAL	1	03/09/99 03/09/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	REN001D	1	OBTAIN	2	08/11/99 08/11/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	REN001D	1	RETRIEVAL	2	08/11/99 08/11/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	REN002D	1	OBTAIN	2	08/11/99 08/11/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	REN002D	1	RETRIEVAL	2	08/11/99 08/11/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	REN003D	1	OBTAIN	2	03/30/99 03/30/99
EMPSCHM	1	EMPSSLR	EMP-DEMO-REGION	REN003D	1	RETRIEVAL	2	03/30/99 03/30/99
EMPSCHM	1	EMPSSLR	ORG-DEMO-REGION	LRTD01D	1	RETRIEVAL	2	03/29/99 03/29/99
EMPSCHM	1	EMPSSLR	ORG-DEMO-REGION	LRTD01D	1	UPDATE	1	03/29/99 03/29/99
EMPSCHM	1	EMPSSLR	ORG-DEMO-REGION	OTFT03D	1	RETRIEVAL	1	03/09/99 03/09/99
EMPSCHM	1	EMPSSLR	ORG-DEMO-REGION	REN001D	1	RETRIEVAL	2	08/11/99 08/11/99
EMPSCHM	1	EMPSSLR	ORG-DEMO-REGION	REN002D	1	RETRIEVAL	2	08/11/99 08/11/99
EMPSCHM	1	EMPSSLR	ORG-DEMO-REGION	REN003D	1	RETRIEVAL	2	03/30/99 03/30/99
EMPSCHM	1	EMPSSLR1	EMP-DEMO-REGION	LRFC01D	1	RETRIEVAL	1	03/30/99 03/30/99
EMPSCHM	1	EMPSSLR1	EMP-DEMO-REGION	LRFC02D	1	RETRIEVAL	1	03/30/99 03/30/99
EMPSCHM	1	EMPSSLR1	ORG-DEMO-REGION	LRFC01D	1	RETRIEVAL	1	03/30/99 03/30/99
EMPSCHM	1	EMPSSLR1	ORG-DEMO-REGION	LRFC02D	1	RETRIEVAL	1	03/30/99 03/30/99
EMPSCHM	1	EMPSS01	EMP-DEMO-REGION	ADMI01D	1	OBTAIN	2	08/13/99 08/13/99

Figure 2-47. Sample DREPORT 029

**Field descriptions:** A description of the fields in the sample report follows:

**SCHEMA/VER**

Identifies the name and version number of the schema that contains the area referenced by the program.

**SUBSCHEMA**

Identifies the subschema in which the area exists.

**AREA**

Identifies the area associated with the named program.

**PROGRAM/VER**

Identifies the name and version number of the program using this area.

**ACTIVITY**

Identifies the DML command issued against this area by the named program.

**TIMES**

Indicates the number of times the DML command is issued against the area within the named program.

**DATE COMPILED/CREATED**

Identifies the date this program was last compiled and the date the program occurrence was defined to the dictionary.

### 2.10.11 Element/Program Cross-Reference Report (DREPORT 030)

**Contents:** The Element/Program Cross-Reference report lists all element/program relationships defined to the dictionary; all elements associated with a record that is referenced by a program are listed. Element/record and record/area relationships are defined automatically for elements, records, and areas stored in a CA-IDMS/DB database. For non-database files, element/record and record/file relationships are defined through the DDDL syntax for elements.

REPORT NO. 30		DATA DICTIONARY REPORTER REL 15.0				09/08/99		PAGE 1	
DREPORT 030		ELEMENT/PROGRAM CROSS REFERENCE REPORT							
*****									
PRIMARY ELEMENT NAME		PROGRAM NAME		USAGE		PROGRAM ELEMENT NAME			
*****									
ACTIVE		VER	1	ADMI01D	VER	1	OBTAIN	ACTIVE-0415	
		VER	1	ADMI01D	VER	1	RETURN	ACTIVE-0415	
		VER	100	ADMS01D	VER	1	OBTAIN	ACTIVE-0415	
		VER	100	ADMS01D	VER	1	RETURN	ACTIVE-0415	
		VER	100	ADMS02D	VER	1	OBTAIN	ACTIVE-0415	
		VER	100	ADMS02D	VER	1	PROT UPD	ACTIVE-0415	
		VER	100	ADMS02D	VER	1	RETURN	ACTIVE-0415	

Figure 2-48. Sample DREPORT 030

**Field descriptions:** A description of the fields in the sample report follows:

**PRIMARY ELEMENT NAME**

Identifies the primary name and version number of an element associated with a record used by a program.

**PROGRAM NAME**

Identifies the name and version number of the program that accesses the record that contains the element.

**USAGE**

Identifies the usage mode of the area associated with the element. If the element/program relationship is established through a CA-IDMS/DB area, the READY mode of the CA-IDMS/DB area is displayed. Possible values for the area READY mode are UPDATE, PROTECTED RETRIEVAL, EXCLUSIVE RETRIEVAL, and EXCLUSIVE UPDATE. If the element/program relationship is defined through a non-database file, the OPEN mode of the non-database file is displayed. Possible values for the file OPEN mode are INPUT, OUTPUT, and I-O. Programs that potentially change or reference an element can be identified by the OPEN mode of the file or by the READY mode of the area.

**PROGRAM ELEMENT NAME**

Identifies the element name used by the program to reference the element. The name can be the primary element name or an element synonym name.

## 2.11 Special-purpose report modules

**Purpose:** There are four special-purpose report modules (DREPORTs 00, 050, 051, and 052). These report modules perform specific functions such as providing housekeeping parameters for the other reports, listing dictionary level numbers, punching module source text to cards, or writing source text to disk. DREPORT 050 is the only special-purpose report that produces printed output.

**Summary table:** The table below lists the four special-purpose report modules in order of presentation in this section.

DREPORT Module	DREPORT Name
00	Comment/Nest Resolution
050	Level Number Report
051	Module Text to Card Utility(1)
052	Module Text to Output File Utility(1)
(1) DREPORTs 051 and 052 must be run alone.	

### 2.11.1 Comment/Nest Resolution Report Module (DREPORT 00)

**Contents:** The comment/nest resolution report module performs internal housekeeping functions whenever a dictionary, DC/UCF system, CA-ADS, or catalog report module is requested. CA-CULPRIT parameters for DREPORT 00 are copied the first time a DREPORT=, CREPORT=, or AREPORT= request parameter is encountered in the input stream. There is no printed output for this report module.

### 2.11.2 Level Number Report (DREPORT 050)

**Contents:** The level number report lists the current values of all level numbers established in the dictionary. Level numbers are assigned by the dictionary when the element is included in a record. Up to 48 levels of data elements can be established by using the LEVEL NUMBERS ARE clause of the DDDL SET OPTIONS statement.

The report below shows sample output for DREPORT 050. This report lists the 48 level numbers and their current values.



REPORT NO. 50 DREPORT 050	DATA DICTIONARY REPORTER REL 15.0 LEVEL NUMBER REPORT	09/08/99	PAGE 1
*****			
LEVEL NUMBERS	----	D A T E	----
*****	UPDATED	CREATED	*****
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	08/18/99	01/08/99	
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49			

Figure 2-49. Sample DREPORT 050

**Field descriptions:** A description of the fields in the sample report follows:

**LEVEL NUMBERS**

Identifies the current values of level numbers within the dictionary. The first 24 levels are listed on line 1; the second 24 levels are listed on line 2.

**DATE UPDATED/CREATED**

Identifies the date the dictionary was last updated and the date the dictionary was defined.

### 2.11.3 Module Text to Card Utility (DREPORT 051)

**Purpose:** The Module Text to Card utility lets you punch the module source code of a specified module to cards. The DREPORT and KEY parameters used to punch module DREPORT 054 to card are:

```
DREPORT=051
KEY MOD-NAME-067 'DREPORT 054'
```

**How to run DREPORT 051:** To run DREPORT 051, include the following specification in the JCL:

■ **For OS/390 systems:**

```
//SYSPCH DD SYSOUT=B,DCB=BLKSIZE=80
```

■ **For VSE/ESA systems:**

```
// ASSGN SYSPCH,X'ccc'
```

---

ccc	device assignment (channel and unit) for punched output
-----	---------------------------------------------------------

---

■ **For VM/ESA systems:**

```
FILEDEF SYSPCH DISK syspch output a
```

---

syspch output a	file identifier of the card-image output file
-----------------	-----------------------------------------------

---

■ **For BS2000/OSD systems:**

---

```

/CREATE-FILE F-NAME=temp.punchcard,(SPACE=(primary,secondary)),      -
/  SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYSPCH,F-NAME=temp.punchcard

```

---

temp.punchcard	file name of punched card output file
primary	primary space allocation
secondary	secondary space allocation

---

After the CA-CULPRIT job runs, you must use the BS2000/OSD PUNCH user command to punch the temp.punchcard file.

**How to request card-image listing:** No *printed* output is produced by this report other than an accounting of the number of records written. To request a card-image listing of the module instead of punched cards, include the following JCL specification:

■ **For OS/390 systems:**

```
//SYSPCH DD SYSOUT=A
```

■ **For VSE/ESA systems:**

```
// ASSGN SYSPCH,X'ppp'
```

---

ppp	printer device assignment
-----	---------------------------

---

■ **For VM/ESA systems:**

```
FILEDEF SYSPCH PRINTER
```

■ **For BS2000/OSD systems:**

```

/CREATE-FILE F-NAME=temp.lstf,(SPACE=(primary,secondary)),      -
/  SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYSPCH,F-NAME=temp.lstf

```

---

temp.lstf	file name of printed output file
primary	primary space allocation
secondary	secondary space allocation

---

After the CA-CULPRIT job runs, you must use the BS2000/OSD PRINT user command to print the temp.lstf file.

## 2.11.4 Module Text to File Utility (DREPORT 052)

**Purpose:** The module text to output file utility (DREPORT 052) lets you output module source code to a disk file. The DREPORT and KEY parameters used to output module DRPT054 to file are shown below:

```

DREPORT=052
KEY MOD-NAME=067 'DREPORT 054'

```

**How to run DREPORT 052:** To run DREPORT 052, add the following specification to the JCL:

■ **For OS/390 files:**

```
//SYS020 DD DSN=user.textfile,DISP=(NEW,CATLG),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=320),
//          UNIT=disk,VOL=SER=nnnnnn
```

---

user.textfile	data set name of the output file
---------------	----------------------------------

---

disk	symbolic device name of disk
------	------------------------------

---

nnnnnn	volume serial number of disk
--------	------------------------------

---

■ **For VSE/ESA tape files:**

```
// TLBL  SYS020,'user.text'
// ASSGN SYS020,TAPE,VOL=nnnnnn
```

---

user.text	file-id of tape file
-----------	----------------------

---

nnnnnn	tape volume serial number
--------	---------------------------

---

■ **For VSE/ESA disk files:**

```
// DLBL  SYS020,'user.text'
// EXTENT SYS020,nnnnnn
// ASSGN SYS020,DISK,VOL=nnnnnn,SHR
```

---

user.text	file-id of disk file
-----------	----------------------

---

nnnnnn	volume serial number of the disk file
--------	---------------------------------------

---

■ **For VM/ESA files:**

```
FILEDEF SYS020 DISK nonprint file a (RECFM FB LRECL 80 BLKSIZE 320
```

---

nonprint file a	filename, filetype, and filemode of the nonprint/nonpunch output file
-----------------	--------------------------------------------------------------------------

---

■ **For BS2000/OSD files:**

```
/CREATE-FILE F-NAME=user.nonprint,(SPACE=(primary,secondary)), -
/ SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYS020,F-NAME=user.nonprint
```

---

user.nonprint	file name for nonprint/nonpunch output file
---------------	---------------------------------------------

---

primary	primary space allocation
---------	--------------------------

---

secondary	secondary space allocation
-----------	----------------------------

---



## Chapter 3. CA-ICMS Catalog Reports — DREPORTS

---

3.1 Purpose	3-3
3.2 Information stored in DDLDML area of dictionary	3-4
3.3 Uses for catalog reports	3-5
3.4 Summary of catalog reports	3-6
3.5 Producing catalog reports	3-7
3.5.1 Syntax	3-7
3.5.2 Parameters	3-7
3.5.3 Examples	3-8
3.6 DREPORT 090 — Catalog Summary Report	3-9
3.6.1 Contents	3-9
3.6.2 Field descriptions	3-9
3.7 DREPORT 091 — Catalog Detail Report	3-11
3.7.1 Contents	3-11
3.7.2 Field descriptions	3-12
3.8 DREPORT 092 — Group Detail Report	3-14
3.8.1 Contents	3-14
3.8.2 Field descriptions	3-14
3.9 DREPORT 093 — User Detail Report	3-16
3.9.1 Contents	3-16
3.9.2 Field descriptions	3-16
3.10 DREPORT 094 — Folder Detail Report	3-19
3.10.1 Contents	3-19
3.10.2 Field descriptions	3-19
3.11 DREPORT 095 — Object Detail Report	3-21
3.11.1 Contents	3-21
3.11.2 Field descriptions	3-21
3.12 DREPORT 096 — Catalog Summary Key Report	3-23
3.12.1 Contents	3-23
3.12.2 Field descriptions	3-23
3.13 DREPORT 097 — Catalog Detail Key Report	3-24
3.13.1 Contents	3-24
3.13.2 Field descriptions	3-24



## 3.1 Purpose

The catalog is a directory of information used by a number of CA-ICMS products, including the Automatic System Facility (ASF) and the Information Center Management System. Eight standard catalog reports provide information about the contents of the catalog. These catalog reports are a subset of the standard DREPORTs.

## 3.2 Information stored in DDLDML area of dictionary

Catalog-related records are stored in the DDLDML area of the dictionary. Information displayed in the fields of the catalog reports is taken from the corresponding fields of the dictionary records.

►► For more information about the structure of the catalog records and how catalog records are defined to the data dictionary, see the &U\$DDR..



## 3.3 Uses for catalog reports

Catalog reports can:

- Help the information center administrator monitor the contents of the dictionary and the catalog structure
- Help users monitor their own private data

## 3.4 Summary of catalog reports

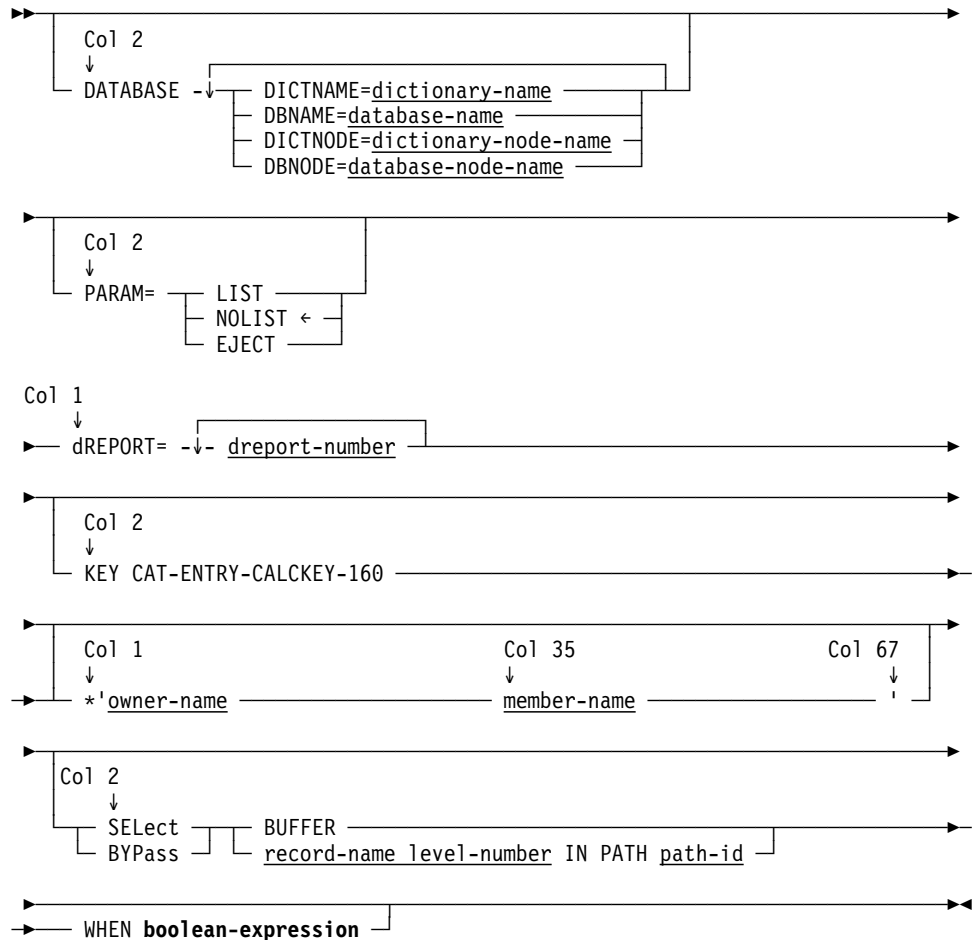
The table below lists the catalog reports in order of presentation in this chapter:

<b>DREPORT Module</b>	<b>DREPORT Name</b>
090	Catalog Summary Report
091	Catalog Detail Report
092	Group Detail Report
093	User Detail Report
094	Folder Detail Report
095	Object Detail Report
096	Catalog Summary Key Report
097	Catalog Detail Key Report

## 3.5 Producing catalog reports

A catalog report is produced by submitting a job that includes the standard JCL for CA-CULPRIT report writers and report-specific control statements. The JCL for OS/390, VSE/ESA, VM/ESA, and BS2000/OSD operating systems is shown in Appendixes A through D, respectively.

### 3.5.1 Syntax



### 3.5.2 Parameters

General syntax rules for DREPORTs are presented in Chapter 1, "Introduction." Syntax rules specific to the catalog reports are described below:

#### **dREPORT=dreport-number**

Identifies the catalog report that you want to run. The D of DREPORT is optional. If D is specified, DREPORT must begin in column 1; if D is not specified, REPORT must begin in column 2.

*Dreport-number* is the 3-digit report number. Catalog reports are assigned numbers 090 through 097. Leading zeros can be omitted.

Multiple reports can be requested in the same job run, except for the key reports (DREPORTs 096 and 097), which must be run alone.

#### **KEY CAT-ENTRY-CALCKEY-160**

(For DREPORTs 096 and 097 only) Identifies the report as a key report; code the literal starting in column 2.

#### **'owner-name member-name'**

Specifies the owner and entity name of the selected entity. This field must be exactly 64 characters long and must be enclosed in quotation marks.

*Owner-name* identifies either a user or the catalog (CORP); it can be up to 32 characters long. If it is less than 32 characters, pad the name with blanks so that the member name starts at character 33 of the literal (that is, within the quotation marks).

*Member-name* specifies the selected entity. *Member-name* can be up to 32 characters long; it must begin in column 33 of the literal. If *member-name* is less than 32 characters, pad the entry with blanks so that the entry is exactly 32 characters long. The closing quotation mark immediately follows *member-name*. (See Example 2 below.)

### **3.5.3 Examples**

**Example 1:** These control statements can be used to run a summary catalog report. The report modules used to run the report are in the default dictionary; the data for the report is taken from the ASFDICT dictionary.

```
DATABASE DBNAME=ASFDICT
DREPORT=90
```

**Example 2:** These control statements can be used to report on all entities whose path includes CORP.FINANCIAL ANALYSIS DATA. The asterisk is the CULPRIT continuation character. Note that CORP begins in column 3 and FINANCIAL ANALYSIS DATA begins in column 35, the 33rd column of the literal. The closing quotation mark is in column 67.

The report modules used to run this report are in the CULPDICT dictionary; data for the report is taken from the ASFDICT dictionary.

```
DATABASE DICTNAME=CULPDICT DBNAME=ASFDICT
DREPORT=097
KEY CAT-ENTRY-CALCKEY-160
*'CORP                                FINANCIAL ANALYSIS DATA'
```

## 3.6 DREPORT 090 — Catalog Summary Report

### 3.6.1 Contents

The Catalog Summary report provides information about all entities defined to the catalog, including their entity types and descriptions. The list is presented in hierarchical order to illustrate the relationships between entities. All catalog entities except passkeys are represented in the dictionary as occurrences of the CATENTRY-160 record. Passkeys are represented as occurrences of the CATPASSKEY-162 record.

REPORT NO. 90 DREPORT 090		DATA DICTIONARY REPORTER REL 15.0 CATALOG SUMMARY		09/08/99 PAGE 1
ENTRY NAME	ENTRY TYPE	TDN	DESCRIPTION	
CORP	CATALOG		DATA DICTIONARY CATALOG	
. DBA GROUP	GROUP		DATABASE ADMINISTRATION GROUP	
. . CARNE01	USER			
. . . CULPRITS	DATATABLE	111	CULPRIT Tests out in ProTesT	
. . CULL DBA	USER		INITIAL DBA GROUP USER	
. . IQA	USER			
. . MJH	USER			
. . . THRUSH	DATATABLE	120		
. . PAGT001	USER			
. . . FOLD	FOLDER			
. . . . SP	DATATABLE	104		
. . . NEWVIEW	DATATABLE	129		
. . . P	DATATABLE	102		
. . . S	DATATABLE	103		
. . . SORTFORM	DATATABLE	106		
. . . SORTLRGE	DATATABLE	105		
. . . SORTSMAL	DATATABLE	107		
. . . SYSTABLE	DATATABLE	113		
. . VQA	USER			
. ENK	USER			
. . DATASF02	DATATABLE	122		

Figure 3-1. Sample DREPORT 090

### 3.6.2 Field descriptions

A description of the fields in the sample report follows:

#### ENTRY NAME

Identifies the names of all catalog entities.

#### ENTRY TYPE

Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

#### TDN

Identifies the table definition number of the object.

**DESCRIPTION**

Identifies the entity description defined to the catalog.

## 3.7 DREPORT 091 — Catalog Detail Report

### 3.7.1 Contents

The Catalog Detail report provides information about all entities defined to the catalog, including their entity types, descriptions, and passkey allocations.

REPORT NO. 91 DREPORT 091		DATA DICTIONARY REPORTER REL 15.0 CATALOG DETAIL	09/08/99	PAGE 1
ENTRY NAME	TYPE	DESCRIPTION		
CORP	CATALOG	DATA DICTIONARY CATALOG OWNER: CREATED 03/10/99 16:56 47.7 IDMSDDDL ACCESSED 00/00/00 00:00 00.0 MODIFIED 00/00/00 00:00 00.0 REDEFINED 00/00/00 00:00 00.0 ALTERED 00/00/00 00:00 00.0  PASSKEYS GIVEN: BROWSE COPY TO USER SQA  PASSKEYS GIVEN: ADD BROWSE COPY CREATE ERASE LIST MANAGEMENT MODIFY REDEFINE TO USER IQA  PASSKEYS GIVEN: ADD BROWSE COPY CREATE ERASE LIST MANAGEMENT MODIFY REDEFINE TO USER PAGT001  PASSKEYS GIVEN: BROWSE CREATE MANAGEMENT TO GROUP DBA GROUP		
. DBA GROUP	GROUP	DATABASE ADMINISTRATION GROUP OWNER: CORP ACCESS: FOUNDATION CREATED 03/10/99 16:56 47.7 IDMSDDDL ACCESSED 00/00/00 00:00 00.0 MODIFIED 00/00/00 00:00 00.0 REDEFINED 00/00/00 00:00 00.0 ALTERED 00/00/00 00:00 00.0  PASSKEYS HELD: MANAGEMENT FOR OBJECT \$UNSTRUCTURED-IDB-OBJECT\$  PASSKEYS HELD: ADD BROWSE COPY CREATE ERASE LIST MANAGEMENT MODIFY REDEFINE FOR FOLDER DBA FOLDER  PASSKEYS HELD: BROWSE CREATE MANAGEMENT FOR CATALOG CORP		
. . CARNE01	USER	OWNER: CORP AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048 ACCESS: CULL DBA CREATED 06/04/99 14:35 08.4 ACCESSED 08/16/99 14:17 42.9 MODIFIED 00/00/00 00:00 00.0 REDEFINED 00/00/00 00:00 00.0 ALTERED 00/00/00 00:00 00.0		
. . . CULPRITS	DATATABLE	CULPRIT Tests out in ProTesT OWNER: CARNE01 OBJECT TYPE: DATATABLE ID: 111 ACCESS: CREATED 04/02/99 12:08 30.4 NBC		

Figure 3-2. Sample DREPORT 091

## 3.7.2 Field descriptions

A description of the fields in the sample report follows:

**ENTRY NAME**

Identifies the entity names in the catalog.

**TYPE**

Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATatable, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

**DESCRIPTION**

Identifies the entity description defined to the catalog.

**OWNER**

Identifies the name of the owner of each entity.

**Date/time stamp**

Information, including the user responsible for the time stamp (IDMSDDL is displayed for catalog foundation entities):

**CREATED**

Specifies the date the entity was defined to the catalog.

**ACCESSED**

Specifies the date the object entity was last accessed or, for a user entity, the date the user last signed on.

**MODIFIED**

Is currently an unused field.

**REDEFINED**

Is currently an unused field.

**ALTERED**

Specifies the date the catalog definition for the entity was last modified.

**Passkey information**

Lists the following:

**PASSKEYS GIVEN**

Identifies the passkeys given to groups or users for access to the listed entity.

**PASSKEYS HELD**

Identifies the passkeys held by the listed entity for access to specific entities.

**OBJECT TYPE/ID**

Identifies the object type and the definition number of each object entity. For a list of object types, see TYPE above.

**ACCESS**

Identifies the type of access control:



**FOUNDATION**

Identifies the entity as a member of the catalog foundation; catalog foundation members cannot be deleted or renamed.

**NO DUPLICATES**

(Users and CORP only) Specifies that duplicate names are not allowed for objects and folders owned by the listed entity.

**NO PROPAGATION**

(Users and CORP only) Specifies that ambiguous associations that involve duplicate names are not allowed.

**ACCESS LOCK**

(Users only) Specifies that the user is restricted from signing on.

**ENCRYPT**

Specifies that the user's password is encrypted.

**Memory requirements**

Identifies memory requirements for each user entity defined to the catalog:

**AFFIL SIZE**

Identifies the number of bytes required to hold the list of groups with which the user is affiliated.

**STACK SIZE**

Identifies the number of bytes required to hold the bill-of-material structure explosion/implosion levels.

**DIRECTORY**

Is currently an unused field.

## 3.8 DREPORT 092 — Group Detail Report

### 3.8.1 Contents

The Group Detail report provides information about all groups defined to the catalog.

REPORT NO. 92 DREPORT 092		DATA DICTIONARY REPORTER REL 15.0			09/08/99 PAGE 1	
GROUP DETAIL						
GROUP NAME	OWNER NAME	TIME STAMP:	DATE	TIME	USER	
DBA GROUP	CORP	CREATED	03/10/99	16:56 47.7	IDMSDDDL	
		ACCESSED	00/00/00	00:00 00.0		
		MODIFIED	00/00/00	00:00 00.0		
		REDEFINED	00/00/00	00:00 00.0		
		ALTERED	00/00/00	00:00 00.0		
DESCRIPTION: DATABASE ADMINISTRATION GROUP						
ACCESS: FOUNDATION						
POSTMASTER	CORP	CREATED	06/25/99	16:03 05.9	IDBCAT	
		ACCESSED	00/00/00	00:00 00.0		
		MODIFIED	00/00/00	00:00 00.0		
		REDEFINED	00/00/00	00:00 00.0		
		ALTERED	00/00/00	00:00 00.0		
DESCRIPTION: FOUNDATION						
ACCESS: FOUNDATION						
QAGROUP	CORP	CREATED	07/13/99	14:31 07.5	PAGT001	
		ACCESSED	00/00/00	00:00 00.0		
		MODIFIED	00/00/00	00:00 00.0		
		REDEFINED	00/00/00	00:00 00.0		
		ALTERED	00/00/00	00:00 00.0		
DESCRIPTION: QUALITY ASSURANCE GROUP						
ACCESS:						

Figure 3-3. Sample DREPORT 092

### 3.8.2 Field descriptions

A description of the fields in the sample report follows:

#### GROUP NAME

Identifies the group defined to the catalog.

#### OWNER NAME

Identifies CORP as the owner of each group.

#### TIME STAMP: DATE/TIME

Identifies the date and time stamps for the group:

##### CREATED

Specifies the date the group definition was defined to the catalog.

##### ACCESSED

Is used for object and user entities only.

##### MODIFIED

Is currently an unused field.

**REDEFINED**

Is currently an unused field.

**ALTERED**

Specifies the date the catalog definition for the group was last modified.

**USER**

Identifies the user responsible for the time stamp. (IDMSDDDL is displayed for DBA GROUP.)

**DESCRIPTION**

Displays the group description defined to the catalog.

**ACCESS**

Indicates whether the group is a member of the catalog foundation (FOUNDATION). Catalog foundation members cannot be renamed or deleted.

## 3.9 DREPORT 093 — User Detail Report

### 3.9.1 Contents

The User Detail report provides information about all user occurrences defined to the catalog, including date/time stamps for the user and the amount of memory required by the user.

REPORT NO. 93 DREPORT 093		DATA DICTIONARY REPORTER REL 15.0 USER DETAIL				09/08/99 PAGE 1		
USER NAME		OWNER NAME		TIME STAMP:		DATE	TIME	USER
CARNE01		CORP		CREATED		06/04/99	14:35 08.4	CULL DBA
				SIGNED ON		08/16/99	14:17 42.9	
AFFIL SIZE: 160		STACK SIZE: 64		MODIFIED		00/00/00	00:00 00.0	
				REDEFINED		00/00/00	00:00 00.0	
				ALTERED		00/00/00	00:00 00.0	
DESCRIPTION: ACCESS:								
CULL DBA		CORP		CREATED		03/10/99	16:56 47.7	IDMSDDDL
				SIGNED ON		07/22/99	15:34 05.6	
AFFIL SIZE: 32		STACK SIZE: 128		MODIFIED		00/00/00	00:00 00.0	
				REDEFINED		00/00/00	00:00 00.0	
				ALTERED		00/00/00	00:00 00.0	
DESCRIPTION: INITIAL DBA GROUP USER ACCESS:								
ENK		CORP		CREATED		05/03/99	15:48 46.0	IDBCAT
				SIGNED ON		06/18/99	10:51 35.8	
AFFIL SIZE: 160		STACK SIZE: 64		MODIFIED		00/00/00	00:00 00.0	
				REDEFINED		00/00/00	00:00 00.0	
				ALTERED		00/00/00	00:00 00.0	
DESCRIPTION: ACCESS:								
HANEL01		CORP		CREATED		07/02/99	15:09 17.6	IDBCAT
				SIGNED ON		07/02/99	15:10 31.2	
AFFIL SIZE: 160		STACK SIZE: 64		MODIFIED		00/00/00	00:00 00.0	
				REDEFINED		00/00/00	00:00 00.0	
				ALTERED		00/00/00	00:00 00.0	
DESCRIPTION: ACCESS:								
IDBSYSTEM		CORP		CREATED		03/12/99	15:54 21.8	IDBCAT
				SIGNED ON		00/00/00	00:00 00.0	
AFFIL SIZE: 160		STACK SIZE: 64		MODIFIED		00/00/00	00:00 00.0	
				REDEFINED		00/00/00	00:00 00.0	
				ALTERED		00/00/00	00:00 00.0	
DESCRIPTION: ACCESS:								
FOUNDATION				ACCESS LOCK				

Figure 3-4. Sample DREPORT 093

### 3.9.2 Field descriptions

A description of the fields in the sample report follows:

#### USER NAME

Identifies the users defined to the catalog.

#### OWNER NAME

Identifies CORP as the owner of each user.

**TIME STAMP: DATE/TIME**

Identifies the date and time stamps for the user:

**CREATED**

Specifies the date the user was defined to the catalog.

**SIGNED ON**

Specifies the date the user last signed on to ASF or IDB.

**MODIFIED**

Is currently an unused field.

**REDEFINED**

Is currently an unused field.

**ALTERED**

Specifies the date the catalog definition for the user was last modified.

**USER**

Identifies the user responsible for the time stamp (IDMSDDDL is displayed for CULL DBA).

**Memory requirements**

Identifies memory requirements for the user:

**AFFIL SIZE**

Indicates the number of bytes required to hold the list of groups with which the user is affiliated.

**STACK SIZE**

Indicates the number of bytes required to hold the bill-of-material structure explosion/implosion levels.

**DIRECTORY**

Is currently an unused field.

**DESCRIPTION**

Displays the user description defined to the catalog.

**ACCESS**

Identifies the access control specified for the user:

**FOUNDATION**

Identifies the user as a member of the catalog foundation; catalog foundation members cannot be deleted or renamed.

**NO DUPLICATES**

Indicates that duplicate names are not allowed for objects and folders owned by the user.

**NO PROPAGATION**

Indicates that ambiguous associations involving duplicate names are not allowed.

**ACCESS LOCK**

Indicates that the user is restricted from signing on.

**ENCRYPT**

Specifies that the user's password is encrypted.

## 3.10 DREPORT 094 — Folder Detail Report

### 3.10.1 Contents

The Folder Detail report provides information about all folders defined to the catalog.

REPORT NO. 94		DATA DICTIONARY REPORTER REL 15.0			09/08/99 PAGE 1	
DREPORT 094		FOLDER DETAIL				
FOLDER NAME	OWNER NAME	TIME STAMP:	DATE	TIME	USER	
DBA FOLDER	IDBSYSTEM	CREATED	03/12/99	15:54 22.0	IDBCAT	
		ACCESSED	00/00/00	00:00 00.0		
		MODIFIED	00/00/00	00:00 00.0		
		REDEFINED	00/00/00	00:00 00.0		
		ALTERED	00/00/00	00:00 00.0		
DESCRIPTION: ACCESS: FOUNDATION						
FOLD	PAGT001	CREATED	07/27/99	15:07 53.9	PAGT001	
		ACCESSED	00/00/00	00:00 00.0		
		MODIFIED	00/00/00	00:00 00.0		
		REDEFINED	00/00/00	00:00 00.0		
		ALTERED	00/00/00	00:00 00.0		
DESCRIPTION: ACCESS:						
PRIVATE FOLDER	IDBSYSTEM	CREATED	03/17/99	14:49 21.2	IDBCAT	
		ACCESSED	00/00/00	00:00 00.0		
		MODIFIED	00/00/00	00:00 00.0		
		REDEFINED	00/00/00	00:00 00.0		
		ALTERED	00/00/00	00:00 00.0		
DESCRIPTION: ACCESS: FOUNDATION						

Figure 3-5. Sample DREPORT 094

### 3.10.2 Field descriptions

A description of the fields in the sample report follows:

#### **FOLDER NAME**

Identifies the folders defined to the catalog.

#### **OWNER NAME**

Identifies the owner of each folder.

#### **TIME STAMP: DATE/TIME**

Identifies date and time stamps for the folder:

#### **CREATED**

Specifies the date the folder definition was created in the catalog.

#### **ACCESSED**

Is used for object and user entities only.

#### **MODIFIED**

Is currently an unused field.

**REDEFINED**

Is currently an unused field.

**ALTERED**

Specifies the date the catalog definition for the folder was last modified.

**USER**

Identifies the user responsible for the time stamp.

**DESCRIPTION**

Displays the folder description defined to the catalog.

**ACCESS**

Indicates whether the folder is a member of the catalog foundation (FOUNDATION). Catalog foundation members cannot be deleted or renamed.



## 3.11 DREPORT 095 — Object Detail Report

### 3.11.1 Contents

The Object Detail report provides information about all objects defined to the catalog.

REPORT NO. 95 DREPORT 095		DATA DICTIONARY REPORTER		REL 15.0	09/08/99 PAGE 1	
OBJECT NAME		OWNER NAME	TIME STAMP:		DATE	TIME USER
\$OBJECT-SECURITY-NAME-TABLE\$		IDBSYSTEM	CREATED	03/12/99	15:56	42.2 CULL DBA
			ACCESSED	05/14/99	17:44	53.6 CULL DBA
OBJECT TYPE: DATATABLE		ID: 21	MODIFIED	00/00/00	00:00	00.0 CULL DBA
			REDEFINED	03/12/99	15:57	04.8 CULL DBA
			ALTERED	00/00/00	00:00	00.0
DESCRIPTION:						
ACCESS:						
\$SECURITY-RUNTIME-TABLE\$		IDBSYSTEM	CREATED	03/12/99	15:54	22.0 CULL DBA
			ACCESSED	05/14/99	17:44	53.7 CULL DBA
OBJECT TYPE: DATATABLE		ID: 20	MODIFIED	00/00/00	00:00	00.0 CULL DBA
			REDEFINED	03/12/99	15:54	47.9 CULL DBA
			ALTERED	00/00/00	00:00	00.0
DESCRIPTION:						
ACCESS:						
\$SRT-OST-CROSS-REFERENCE\$		IDBSYSTEM	CREATED	05/14/99	17:44	46.8 CULL DBA
			ACCESSED	05/17/99	22:11	10.0 CULL DBA
OBJECT TYPE: DATATABLE		ID: 22	MODIFIED	00/00/00	00:00	00.0 CULL DBA
			REDEFINED	05/14/99	17:44	53.9 CULL DBA
			ALTERED	00/00/00	00:00	00.0
DESCRIPTION:						
ACCESS:						
\$UNSTRUCTURED-IDB-OBJECT\$		IDBSYSTEM	CREATED	06/25/99	16:05	02.8 IDBCAT
			ACCESSED	07/30/99	11:59	36.5 MJH
OBJECT TYPE: DATATABLE		ID: 101	MODIFIED	00/00/00	00:00	00.0 PAGT001
			REDEFINED	06/25/99	16:05	04.4
			ALTERED	00/00/00	00:00	00.0
DESCRIPTION:						
ACCESS:						
CULPRITS		CARNE01	CREATED	04/02/99	12:08	30.4 NBC
			ACCESSED	06/04/99	14:27	26.0 CARNE01
OBJECT TYPE: DATATABLE		ID: 111	MODIFIED	00/00/00	00:00	00.0 NBC
			REDEFINED	04/02/99	12:08	31.4 NBC
			ALTERED	04/02/99	12:08	55.3 NBC
DESCRIPTION: CULPRIT Tests out in ProTesT						
ACCESS:						

Figure 3-6. Sample DREPORT 095

### 3.11.2 Field descriptions

A description of the fields in the sample report follows:

#### OBJECT NAME

Identifies the objects in the catalog.

#### OWNER NAME

Identifies the owner of each object.

**TIME STAMP: DATE/TIME**

Identifies date and time stamps for the object:

**CREATED**

Specifies the date the object definition was defined to the catalog.

**ACCESSED**

Specifies the date the object was last accessed.

**MODIFIED**

Is currently an unused field.

**REDEFINED**

Is currently an unused field.

**ALTERED**

Specifies the date the catalog definition for the object was last modified.

**USER**

Identifies the user responsible for the time stamp.

**OBJECT TYPE**

Identifies the type of object being described as GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

**ID**

Identifies the table definition number.

**DESCRIPTION**

Displays the object description defined to the catalog.

**ACCESS**

Indicates whether the object is a member of the catalog foundation (FOUNDATION). Catalog foundation members cannot be renamed or deleted.

## 3.12 DREPORT 096 — Catalog Summary Key Report

### 3.12.1 Contents

The Catalog Summary Key report provides information about selected entities defined to the catalog. The selection is based on a key, which consists of an owner name and entity name. Any catalog entity except the passkey entity can be used as the key.

The figure below shows sample output for DREPORT 096. The DREPORT and KEY parameters used to create the sample report are:

```
DREPORT=096
KEY CAT-ENTRY-CALCKEY-160
*'MJH                                BIRDS'
```

REPORT NO. 96		DATA DICTIONARY REPORTER REL 15.0		09/08/99 PAGE 1	
DREPORT 096		CATALOG SUMMARY			
ENTRY NAME	ENTRY TYPE	TDN	DESCRIPTION		
MJH	USER		OWNER: CORP		
. THRUSH	DATATABLE	120			

Figure 3-7. Sample DREPORT 096

### 3.12.2 Field descriptions

A description of the fields in the sample report follows:

**ENTRY NAME**

Identifies the entities being described.

**ENTRY TYPE**

Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, or SYNONYM.

**TDN**

Identifies the table definition number of the object.

**DESCRIPTION**

Identifies the entity description defined to the catalog.

**OWNER**

Identifies the owner of the catalog entity.

## 3.13 DREPORT 097 — Catalog Detail Key Report

### 3.13.1 Contents

The Catalog Detail Key report provides detailed information about selected entities, including their entity types, descriptions, and passkey allocations. The selection is made based on a key, which includes an owner name and member name. Any catalog entity except the passkey entity can be used as the key.

The figure below shows sample output for DREPORT 097. The DREPORT and KEY parameters used to create this sample report are:

```
DREPORT=097
KEY CAT-ENTRY-CALCKEY-160
*'CORP                                CARNE01                                '
```

REPORT NO. 97 DREPORT 097	DATA DICTIONARY REPORTER REL 15.0 CATALOG DETAIL	09/08/99	PAGE 1
ENTRY NAME	TYPE	DESCRIPTION	
CARNE01	USER	OWNER: CORP AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048 ACCESS: FOUNDATION NO DUPLICATES NO PROPAGATION ACCESS LOCK ENCRYPT CREATED 06/04/99 14:35 08.4 CULL DBA ACCESSED 08/16/99 14:17 42.9 MODIFIED 00/00/00 00:00 00.0 REDEFINED 00/00/00 00:00 00.0 ALTERED 00/00/00 00:00 00.0	
. CULPRITS	DATATABLE	CULPRIT Tests out in ProTest OWNER: CARNE01 OBJECT TYPE: DATATABLE ID: 111 ACCESS: CREATED 04/02/99 12:08 30.4 NBC ACCESSED 06/04/99 14:27 26.0 CARNE01 MODIFIED 00/00/00 00:00 00.0 REDEFINED 04/02/99 12:08 31.4 NBC ALTERED 04/02/99 12:08 55.3 NBC	

Figure 3-8. Sample DREPORT 097

### 3.13.2 Field descriptions

A description of the fields in the sample report follows:

#### ENTRY NAME

Identifies the entity being described.

#### TYPE

Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

**DESCRIPTION**

Displays the entity description defined to the catalog.

**OWNER**

Identifies the owner of each entity.

**Memory requirements**

Identifies memory requirements for the user:

**AFFIL SIZE**

Indicates the number of bytes required to hold the list of groups with which the user is affiliated.

**STACK SIZE**

Indicates the number of bytes required to hold the bill-of-material structure explosion/implosion levels.

**DIRECTORY**

Is currently an unused field.

**ACCESS**

Indicates the access control specified for the entity:

**FOUNDATION**

Identifies the entity as a member of the catalog foundation; members of the catalog foundation cannot be renamed or deleted.

**NO DUPLICATES**

(Users and CORP only) Indicates that duplicate names are not allowed for objects and folders owned by the listed entity.

**NO PROPAGATION**

(Users and CORP only) Indicates that ambiguous associations involving duplicate names are not allowed.

**ACCESS LOCK**

(Users only) Indicates that the user is restricted from signing on.

**ENCRYPT**

Specifies that the user's password is encrypted.

**Date/time stamp**

Information is listed, including the user responsible for the time stamp (IDMSDDDL is displayed for catalog entities):

**CREATED**

Specifies the date the object was defined to the catalog.

**ACCESSED**

Specifies the date the object was last accessed.

**MODIFIED**

Is currently an unused field.

**REDEFINED**

Is currently an unused field.

**ALTERED**

Specifies the date the catalog definition for the entity was last modified.

**Passkey information**

Lists the following:

**PASSKEYS GIVEN**

Identifies the passkeys given to groups or users for access to the listed entity.

**PASSKEYS HELD**

Identifies the passkeys held by the listed entity for access to specific entities.

## Chapter 4. DC/UCF System Reports — CREPORTS

---

4.1 Overview	4-3
4.2 Uses for DC/UCF system reports	4-4
4.3 Summary of CREPORTs	4-5
4.4 Producing DC/UCF system reports	4-7
4.4.1 Syntax	4-8
4.4.2 Parameters	4-8
4.4.3 Examples	4-8
4.5 CA-ADS Parameter Reports (CREPORTs 040 and 045)	4-10
4.5.1 Contents	4-10
4.5.2 Field descriptions	4-10
4.6 Defined Devices Report (CREPORT 029)	4-13
4.6.1 Contents	4-13
4.6.2 Field descriptions	4-13
4.7 Defined Messages Report (CREPORT 028)	4-14
4.7.1 Contents	4-14
4.7.2 Field descriptions	4-14
4.8 Destination Reports (CREPORTs 007 and 024)	4-16
4.8.1 Contents	4-16
4.8.2 Field descriptions	4-16
4.9 Load Area Report (CREPORT 050)	4-18
4.9.1 Contents	4-18
4.9.2 Field descriptions	4-18
4.10 Mapping Reports (CREPORTs 030 through 035)	4-19
4.10.1 Contents	4-19
4.10.2 Field descriptions	4-21
4.11 Nodes and Resource Table Reports (CREPORTs 043 and 044)	4-23
4.11.1 Contents	4-23
4.11.2 Field descriptions	4-24
4.12 Module Text to Card Utility (CREPORT 051)	4-25
4.13 Module Text to File Utility (CREPORT 052)	4-27
4.14 Network Description Reports (CREPORTs 001-003, 014-018)	4-29
4.14.1 Contents	4-29
4.14.2 Sample reports	4-29
4.14.3 Field descriptions	4-31
4.15 CA-OLQ Reports (CREPORTs 041 and 046)	4-34
4.15.1 Contents	4-34
4.15.2 Field descriptions	4-34
4.16 Program Description Reports (CREPORTs 004 and 019)	4-37
4.16.1 Contents	4-37
4.16.2 Field descriptions	4-37
4.17 Queue Description Reports (CREPORTs 006, 022, and 023)	4-40
4.17.1 Contents	4-40
4.17.2 Field descriptions	4-40
4.18 Symbol Table Report (CREPORT 053)	4-42
4.18.1 Contents	4-42
4.18.2 Field descriptions	4-42
4.19 System Options Reports (CREPORTs 011 and 025)	4-43

---

4.19.1	Contents	4-43
4.19.2	Field descriptions	4-44
4.20	Task Description Reports (CREPORTs 005, 020, and 021)	4-53
4.20.1	Contents	4-53
4.20.2	Field descriptions	4-53
4.21	Builder codes	4-57



## 4.1 Overview

The CA-IDMS/DC and CA-IDMS/UCF (DC/UCF) system reports describe the dictionary entities used to define the characteristics and components of an executable DC/UCF system. The DC/UCF system reports document information maintained in the DDLML, DDLDCMSG, and DDLDCLOD areas of the dictionary. The records being reported on are defined to the dictionary through various CA-IDMS system software components, such as the Data Dictionary Definition Language (DDDL) compiler and the system generation compiler.

►► For details about the structure of these records and how the records are defined to the dictionary, see the &U\$DDR..

## 4.2 Uses for DC/UCF system reports

DC/UCF system reports can be used to:

- Provide descriptions of executable systems and of systems that have been defined but not generated.

DC/UCF system object reports describe executable systems (that is, systems for which GENERATE has been issued in the system generation compile); these reports access dictionary object records. DC/UCF system source reports describe systems that have been defined to the dictionary but not yet generated; these reports access dictionary source records.

►► For more information about source and object records, see *CA-IDMS System Generation*.

Because the dictionary structure includes source and object records, you can modify the system definitions without affecting the run-time definitions. The system source reports can be used to review the proposed modifications before they are implemented. Once you are satisfied with the new configuration, you can update the run-time definitions by issuing a system generation GENERATE command for any system that you want to update.

- Review the screen/data field relationships for mapping operations.
- Monitor the contents of the load area.
- Monitor the messages defined to the DDLDCMSG area of the dictionary.

## 4.3 Summary of CREPORTs

The DC/UCF system reports are presented in alphabetical order, with the following exceptions:

- The Physical Terminal by Line and Logical Terminal by Physical Terminal reports (CREPORTs 016 and 018) are discussed under 4.14, “Network Description Reports (CREPORTs 001-003, 014-018)” on page 4-29.
- The Listing of Map reports (CREPORTs 032 through 035) are discussed under 4.10, “Mapping Reports (CREPORTs 030 through 035)” on page 4-19.

The table below lists the CREPORTs in the order of presentation in this chapter. For a list of CREPORTs by report number, see Table E-2 on page E-5.

<b>CREPORT Number</b>	<b>CREPORT Name</b>
040	ADS/OnLine Parameters Report (Object)
045	ADS/OnLine Parameters Report (Source)
029	Defined Devices
028	Defined Messages(1)
007	Destination Report (Object)
024	Destination Report (Source)
050	Load Area Report(1)
030	Map Record Indices
031	Map Field Indices
032	Listing of Maps by Panel
033	Listing of Maps
034	Listing of Maps by Record Name
035	Listing of Maps by Element Name
043	Listing of Nodes
044	Listing of Defined Resources
051	Module Text to Card Utility
052	Module Text to File Utility
001	Network Description by Line (Object)
014	Network Description by Line (Source)
002	Network Description by Physical Terminal (Object)

#### 4.3 Summary of CREPORTs

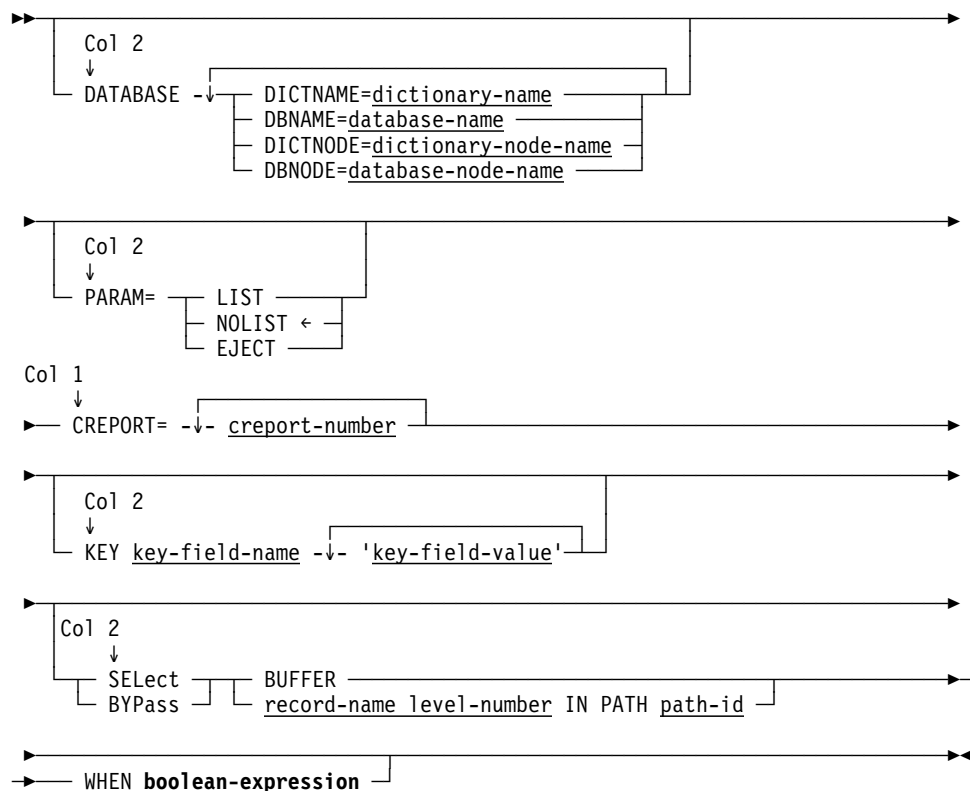
---

<b>CREPORT Number</b>	<b>CREPORT Name</b>
015	Network Description by Physical Terminal (Source)
003	Network Description by Logical Terminal (Object)
017	Network Description by Logical Terminal (Source)
016	Physical Terminals within Line (Source)
018	Logical Terminal by Physical Terminal (Source)
041	OLQ Report (Object)
046	OLQ Report (Source)
004	Program Description Report (Object)
019	Program Description Report (Source)
006	Queue Description Report (Object)
022	Queue Description Report (Source)
023	Queue Description within Task Report (Source)
053	Symbol Table Report
011	System Options Report (Object)
025	System Options Report (Source)
005	Task Description Report (Object)
020	Task Description Report (Source)
021	Task Description within Program Report (Source)
(1) Reports 28 and 50 must be run alone.	

## 4.4 Producing DC/UCF system reports

A DC/UCF system report is produced by submitting a batch job that includes the standard JCL for the CA-CULPRIT report writer and report-specific control statements. JCL for OS/390, VSE/ESA, VM/ESA, and BS2000/OSD operating systems is shown in Appendixes A through D, respectively.

### 4.4.1 Syntax



### 4.4.2 Parameters

General syntax rules for the CA-IDMS reports are described in detail in Chapter 1, "Introduction." Syntax rules specific to the CREPORTs are discussed below:

#### **CREPORT=creport-number**

Identifies the DC/UCF system report being requested. CREPORT must begin in column 1; all other parameters begin in column 2.

*Creport-number* is a 3-digit number that identifies the report module. Leading zeros can be omitted.

With the exception of CREPORTs 028 and 050, which must be run alone, multiple reports can be requested in the same job run.

### 4.4.3 Examples

**Example 1:** These control statements can be used to request CREPORTS 001, 010, and 041. Optionally, you can omit leading zeros for the report modules and/or use a blank or equal sign to separate each report. The report modules used to run the reports are in the CULPDICT dictionary; data for the reports is taken from the DOCUDICT dictionary.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
CREPORT=001,010,041
```

The same reports can be requested by specifying each report on a separate line:

```
CREPORT=1
CREPORT=10
CREPORT=41
```

**Example 2:** These control statements can be used to produce a report on the the task occurrence BYE. The report modules used to produce the report are in the default dictionary; data for the report is taken from the DOCUNWK dictionary.

```
DATABASE DBNAME=DOCUNWK
CREPORT=005
SEL BUFFER WHEN TASK-NAME-023 EQ 'BYE'
```

**Example 3:** These control statements can be used to produce a report on all message occurrences whose id begins with DC301. The report modules used to produce the report are in the CULPDICT dictionary; data for the report is taken from the default dictionary.

```
DATABASE DICTNAME=CULPDICT
CREPORT=005
SEL MSG-KEY-116 WHEN MSG-KEYRED-116 EQ 'DC301'
```

**Example 4:** These control statements can be used to request the three network object reports. A parameter listing is requested for CREPORT 001; parameter listings are not requested for CREPORTs 002 and 003.

```
DATABASE DICTNAME=CULPDICT DBNAME=TESTDICT
PARAM=LIST
CREPORT=1
PARAM=NOLIST
CREPORT=2,3
```

## 4.5 CA-ADS Parameter Reports (CREPORTs 040 and 045)

### 4.5.1 Contents

The CA-ADS parameter reports provide information about the CA-ADS run-time environment. The CA-ADS definitions for each system are represented in the dictionary by the CVGDEFS-142 record, which is a logical extension of the SYS-041 record.

CREPORT 040 describes the CA-ADS environment for executable systems;  
CREPORT 045 describes the CA-ADS environment for systems that have been defined but not generated. The fields and format of the two reports are the same.

REPORT NO. 40 CREPORT 040				CA-IDMS/DC ADS ONLINE REPORT REL 15.0 LISTING OF ADS ONLINE PARAMETERS OBJECT REPORT						09/08/99 PAGE 1		
SYSTEM VERSION 99	AUTO DIALOG	PRIMARY TASK CODE ADS	SECONDARY TASK CODE ADS2	TCF TASK CODE ADS2T	MAXIMUM LINKS 10	MENU IS USER		PRIMARY POOL 4084	SECONDARY POOL 4084	BUILDER CODE R	ACTION CODE	ACTIV LOG YES
	DIAGNOSTIC SCREEN YES	NEWPAGE MAPOUT NO	RESOURCES ARE FIXED	- - AUTOSTATUS ENABLED YES	- - OPT/MAND OPTIONAL	- COBOL ENABLED NO	MOVE - OPT/MAND OPTIONAL	- - - ON/OFF OFF	DIALOG STATISTICS SEL/ALL ALL	- - - INTERVAL 0	- - - RECORD COMPRESSION NO	STORAGE MODE SGNSIZE
	- - FAST	MODE THRESHOLD / OFF	INTERVAL 1	- -	- - - - - OPTIONAL	- - - - - ADS0-STAT-DEF-REC	STATUS DEFINITION RECORD	- - - - - VERSION	- - - - - 1			
SYSTEM VERSION 105	AUTO DIALOG	PRIMARY TASK CODE ADS	SECONDARY TASK CODE ADS2	TCF TASK CODE ADS2T	MAXIMUM LINKS 10	MENU IS USER		PRIMARY POOL 4084	SECONDARY POOL 33000	BUILDER CODE R	ACTION CODE	ACTIV LOG YES
	DIAGNOSTIC SCREEN YES	NEWPAGE MAPOUT NO	RESOURCES ARE FIXED	- - AUTOSTATUS ENABLED YES	- - OPT/MAND OPTIONAL	- COBOL ENABLED NO	MOVE - OPT/MAND OPTIONAL	- - - ON/OFF ON	DIALOG STATISTICS SEL/ALL ALL	- - - INTERVAL 200	- - - RECORD COMPRESSION NO	STORAGE MODE SGNSIZE
	- - FAST	MODE THRESHOLD / OFF	INTERVAL 1	- -	- - - - - OPTIONAL	- - - - - ADS0-STAT-DEF-REC	STATUS DEFINITION RECORD	- - - - - VERSION	- - - - - 1			

Figure 4-1. Sample CREPORT 040

### 4.5.2 Field descriptions

A description of the fields in the sample report follows:

#### SYSTEM VERSION

Identifies the system associated with these CA-ADS parameters.

#### AUTO DIALOG

Identifies the mainline dialog that is executed immediately at run time (the mainline dialog menu screen is bypassed).

#### PRIMARY TASK CODE

Identifies the task code entered by the user to initiate the CA-ADS run-time system. This task code must invoke the program ADSORUN1; the default is ADS.



**SECONDARY TASK CODE**

Identifies the task code that invokes the program ADSOMAIN; the default is ADS2TASK.

**TCF TASK CODE**

Identifies the task code that invokes the program ADSOMAIN while running under the transfer control facility (TCF). The default is ADS2T.

**MAXIMUM LINKS**

Identifies the maximum number of dialog levels that can be defined by each respective CA-ADS application thread. The default is 10.

**MENU IS**

Identifies which dialog names will be displayed on the CA-ADS menu. USER (default) indicates that only those mainline dialog names for which the current user is authorized will be displayed; ALL indicates that all mainline dialog names known to the DC/UCF system will be displayed.

**PRIMARY POOL**

Indicates the size of the primary record buffer. The default is 4000.

**SECONDARY POOL**

Indicates the size of the secondary pool to be allocated from DC storage when the primary pool is full. The default is 2000.

**BUILDER CODE**

Identifies the builder code for the record that defines the CA-ADS parameters. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

**ACTION CODE**

Indicates whether the record that defines the CA-ADS parameters has been updated (U) or deleted (D). Blank indicates that the CA-ADS parameters have not been updated since the last time the system was generated.

**ACTIV LOG**

Indicates whether the activity log for the database commands in a dialog is on (YES) or off (NO).

**DIAGNOSTIC SCREEN**

Indicates whether CA-ADS bypasses display of the Dialog Abort Information screen when the run-time system abandons a dialog. YES indicates that the diagnostic screen will not be displayed; NO indicates that the screen will be displayed.

**NEWPAGE MAPOUT**

Indicates how a mapout is performed when a dialog's map is already displayed as the result of a previous mapout. NO indicates that CA-ADS will transmit only the map's data fields and message field; YES indicates that CA-ADS will always perform a new page mapout.

**RESOURCES ARE**

Specifies whether information from the CA-ADS Terminal Block (OTB) and the variable dialog block (VDB) in the storage pool will be written to the scratch area (DDLDCSCR) across a pseudo-converse.

FIXED indicates that the storage is not relocatable; the buffer pools remain in the storage pool provided that the fast mode threshold has not been exceeded.

RELOCATABLE indicates that the storage is relocatable; the buffer pools are written to scratch across a pseudo-converse and the storage is freed.

**AUTOSTATUS ENABLED**

Indicates whether (YES) or not (NO) the AUTOSTATUS setting for dialog generation sessions is enabled.

**AUTOSTATUS OPT/MAND**

Indicates whether (OPTIONAL) or not (MANDATORY) the application developer is allowed to override the default AUTOSTATUS setting during dialog generation.

**COBOL MOVE ENABLED**

Indicates how CA-ADS moves the result of an arithmetic or assignment command into the target field. YES instructs CA-ADS to use COBOL rules; NO instructs CA-ADS to use CA-ADS rules.

**COBOL MOVE OPT/MAND**

Indicates whether (OPTIONAL) or not (MANDATORY) the application developer is allowed to change the COBOL MOVE setting on a dialog-by-dialog basis.

**DIALOG STATISTICS ON/OFF**

Indicates whether (YES) or not (NO) dialog statistics are collected for CA-ADS dialogs.

**DIALOG STATISTICS SEL/ALL**

Indicates whether statistics are collected for dialogs on a dialog-by-dialog basis (SELECTED) or for all dialogs (ALL).

**DIALOG STATISTICS INTERVAL**

Specifies the interval at which dialog statistics are written to the log file after being accumulated the indicated number of times. 0 indicates that statistics are written to the log file after an application terminates.

**RECORD COMPRESSION**

Indicates whether (YES) or not (NO) record buffer blocks (RBBs) are compressed across a pseudo-converse when they are retained in the storage pool.

**STORAGE MODE**

Indicates how storage for record buffer blocks (RBBs) is to be allocated. SGENSIZE uses the buffer sizes specified in the PRIMARY POOL and SECONDARY POOL parameters of the ADSO system generation statement; CALCULATED uses the calculated size of the RBBs for an application or dialog when allocating storage.

**FAST MODE THRESHOLD / INTERVAL**

Indicates whether (ON) or not (OFF) the CA-ADS runtime system writes record buffer blocks (RBBs) and statistics control blocks to scratch across a pseudo-converse. INTERVAL specifies the size of the fast mode threshold.

**STATUS DEFINITION RECORD**

Specifies the name and version of the status definition record and whether (OPTIONAL) or not (MANDATORY) the application developer is allowed by override the default status definition record specification during dialog generation.

## 4.6 Defined Devices Report (CREPORT 029)

### 4.6.1 Contents

The Defined Devices report provides information about line and physical terminal device types supported by DC/UCF. CREPORT 029 obtains information from the DCDEVICES-127 dictionary record, which is stored at installation. DC/UCF compilers use DCDEVICES-127 record occurrences when handling device-dependent syntax.

REPORT NO. 29 CREPORT 029		CA-IDMS/DC SYSTEM GENERATION REPORT DEFINED DEVICES		REL 15.0	09/08/99 PAGE 1	
DEFINED DEVICE	LOWER PARSE MODULE	LINE/TERMINAL	TERMINAL CODE	DEVICE CODE	ACCESS METHOD	
ASR33	RHDCP06E	TERM	19			
ASYN	RHDCL06E	LINE		6	E	
BSC1	RHDCL0BB	LINE		B	B	
BSC2	RHDCL0CB	LINE		C	B	
BSC3	RHDCL0DB	LINE		D	B	
BULK	RHDCP0LS	TERM	47			
CCI CONSOLE	RHDCL0LS RHDCL04W	LINE LINE		L 4	S W	

Figure 4-2. Sample CREPORT 029

### 4.6.2 Field descriptions

A description of the fields in the sample report follows:

#### **DEFINED DEVICE**

Identifies the device type being described.

#### **LOWER PARSE MODULE**

Identifies the name of the syntax tree used to handle the device.

#### **LINE/TERMINAL**

Indicates whether the device being described is a line or a physical terminal.

#### **TERMINAL CODE**

Identifies the terminal type code of the physical terminal. For a list of the possible values, see the *CA-IDMS DSECT Reference*.

#### **DEVICE CODE and ACCESS METHOD**

Identify the line type code and the line/terminal access method code for each line.

#### **COMMENTS**

Displays any comments associated with the occurrence. This field is not shown in the sample report.

## 4.7 Defined Messages Report (CREPORT 028)

### 4.7.1 Contents

The Defined Messages report provides information about messages that have been defined for the DC/UCF system. Messages are represented in the DDLDCMSG area of the dictionary by the MESSAGE-116 record.

REPORT NO. 28 CREPORT 028	CA-IDMS/DC SYSTEM GENERATION REPORT DEFINED MESSAGES	REL 15.0	09/08/99 PAGE 29
MESSAGE ID: AB030026	BUILDER CODE: D		
SEVERITY: 6	DESTINATIONS:		DESTINATION ID:
MESSAGE TEXT LINE NUMBER 1:			
DATA TYPE NOT RECOGNIZED			
COMMENT NUMBER 100:			
COMMENT NUMBER 100:	MODULE(S) = _____		
COMMENT NUMBER 200:			
COMMENT NUMBER 300:	ADDITIONAL INFORMATION ...		
COMMENT NUMBER 400:			
COMMENT NUMBER 500:			
MESSAGE ID: AB030057	BUILDER CODE: D		
SEVERITY: 6	DESTINATIONS:		DESTINATION ID:
MESSAGE TEXT LINE NUMBER 1:			
UNDETERMINED SYNTAX ERROR IN INSTRUCTION			
COMMENT NUMBER 100:			
COMMENT NUMBER 100:	MODULE(S) = _____		
COMMENT NUMBER 200:			

Figure 4-3. Sample CREPORT 028

### 4.7.2 Field descriptions

A description of the fields in the sample report follows:

#### MESSAGE ID

Specifies the identifier assigned to the message.

#### BUILDER CODE

Identifies the builder code for the record that defines the message. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

#### SEVERITY

Identifies the severity level assigned to the message.

#### DESTINATIONS

Identifies the destinations to which the message line is routed.

#### DESTINATION ID

Identifies the terminal to which the message line is routed when the destination identifier flag is set.

**MESSAGE TEXT and LINE NUMBER**

Identify the text of the message. When multiple lines are defined for a message, each line is listed separately.

**COMMENT NUMBER**

Identifies comments associated with the message.

## 4.8 Destination Reports (CREPORTs 007 and 024)

### 4.8.1 Contents

Destination reports provide information about destination occurrences that have been defined to the dictionary. CREPORT 007 provides destination information for executable systems; these destinations are represented in the dictionary by the DESTLST-027 record. CREPORT 024 provides information for systems that have been defined but not generated; these destinations are represented in the dictionary by the DEST-028 record. The fields and format of the two reports are the same.

REPORT NO. 07 CREPORT 007	CA-IDMS/DC SYSTEM GENERATION REPORT			REL 15.0	09/08/99	PAGE 1
	SYSTEM NAME: DCSYSTEM					
	SYSTEM VERSION: 105					
	DESTINATION REPORT					
DESTINATION	BUILDER	ACTION CODE	VERSION	DISABLED	MEMBER TYPE	
USWSWDPL	R		1	YES	TERMINAL	
TERMINAL NAME						
USWSWDPL						

Figure 4-4. Sample CREPORT 007

### 4.8.2 Field descriptions

A description of the fields in the sample report follows:

#### **SYSTEM NAME and VERSION**

Identify the name and version number of the system associated with the destination being described.

#### **DESTINATION**

Identifies the destination being described.

#### **BUILDER**

Identifies the builder code for the record that defines the destination. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

#### **ACTION CODE**

Indicates whether the destination occurrence has been updated (U) or deleted (D). A blank indicates that the destination has not been changed since that last time the system was generated.

#### **VERSION**

Indicates the version number of the destination occurrence.

#### **DISABLED**

Indicates whether the destination is disabled at system startup.

**MEMBER TYPE**

Indicates whether the destination group is made up of users or logical terminals.

**USER NAME or TERMINAL NAME**

Identifies the users in the destination group. If the destination group is made up of logical terminals, TERMINAL NAME is displayed. If the destination group is made up of users, USER NAME is displayed.

## 4.9 Load Area Report (CREPORT 050)

### 4.9.1 Contents

The Load Area report provides information about load modules. Load modules are stored in the DDLDCLOD area of the dictionary and are represented in the dictionary by the LOADHDR-156 record.

REPORT NO. 50 CREPORT 050		CA-IDMS/DC LOAD AREA REPORT LOAD MODULES				REL 15.0		09/08/99 PAGE 1	
MODULE NAME	VERSION	NUM RLD ENTRIES	ENTRY PT ADDRESS	MODULE LENGTH	COMPILE DATE	COMPILE TIME	DELETE FLAG	MODULE TYPE	
\$ACF@TAT	1	0	0	118	04/15/99	211629	OFF	TABLE	
IDMSCSTB	99	5	0	563	08/16/99	223307	OFF	UNDEFINED	
IDMSCSTB	105	5	0	930	08/16/99	224154	OFF	UNDEFINED	
IDMSCSTB	777	5	0	563	06/08/99	194951	OFF	UNDEFINED	
SQACVSS1	1	121	0	1,640	07/06/99	163609	OFF	SUBSCHEMA	
SQACVSS2	1	84	0	1,176	07/06/99	163612	OFF	SUBSCHEMA	

Figure 4-5. Sample CREPORT 050

### 4.9.2 Field descriptions

A description of the fields in the sample report follows:

#### MODULE NAME and VERSION

Identify the name and version number of the load module being described.

#### NUM RLD ENTRIES

Identifies the number of entries in the relocation dictionary (RLD) for the load module.

#### ENTRY PT ADDRESS

Identifies the entry point address of the load module.

#### MODULE LENGTH

Identifies the length, in bytes, of the object text.

#### COMPILE DATE

Identifies the date the load module was created.

#### COMPILE TIME

Identifies the time the load module was created.

#### DELETE FLAG

Indicates whether the load module has been logically deleted. The flag is set ON when a new module is generated for the load module in use or when a deletion is requested. The module is not eligible for replacement when the flag is OFF.

#### MODULE TYPE

Identifies the type of load module: access module, subschema, map, CA-ADS dialog, edit/code table, or mainline dialog.



## 4.10 Mapping Reports (CREPORTs 030 through 035)

### 4.10.1 Contents

Mapping reports describe the interrelationships among maps, panels, map fields, and panel fields that have been defined for DC/UCF systems. Maps are represented in the dictionary by the MAP-098 record; map fields are represented by the MAPFLD-124 record. The table below summarizes each report:

CREPORT 030	Map Record Indices	Lists map record elements sorted on map name
CREPORT 031	Map Field Indices	Lists map panel-fields sorted on map name
CREPORT 032	Maps by Panel	Lists map occurrences sorted on panel name
CREPORT 033	Maps in alphabetic order	Lists all map occurrences in alphabetic order
CREPORT 034	Maps by Record Name	Lists map occurrences by map record name
CREPORT 035	Maps by Element Name	Lists map occurrences sorted on map element name

REPORT NO. 30 CREPORT 030		CA-IDMS/DC MAPPING REPORT REL 15.0				09/08/99 PAGE 1		
		MAP NAME: ADMI01M						
		MAP VERSION: 1						
		MAP RECORD INDICES						
OLMPF-0003	CURSOR	ALARM NO	UNLOCK YES	RESET YES	MAP DATE 03/02/99	MAP TIME 112042	FIELD COUNT 4	RECORD COUNT 1
	RECORD NAME	INDEX						
	EMPLOYEE	1						

Figure 4-6. Sample CREPORT 030

REPORT NO. 31 CREPORT 031		CA-IDMS/DC MAPPING REPORT REL 15.0				09/08/99 PAGE 1		
		MAP NAME: ADMI01M						
		MAP VERSION: 1						
		MAP FIELD INDICES						
OLMPF-0003	CURSOR	ALARM NO	UNLOCK YES	RESET YES	MAP DATE 03/02/99	MAP TIME 112042	FIELD COUNT 4	RECORD COUNT 1
	FIELD NAME	INDEX						
	EMP-ID-0415	2						
	EMP-NAME-0415	3						

Figure 4-7. Sample CREPORT 031

#### 4.10 Mapping Reports (CREPORTs 030 through 035)

REPORT NO. 32 CREPORT 032		CA-IDMS/DC MAPPING REPORT REL 15.0 LISTING OF MAPS BY PANEL PANEL NAME: ABCD01M OLMPANEL				09/08/99 PAGE 1
		PANEL VERSION: 1 PANEL BUILDER: G				
MAP NAME	MAP VERSION	BUILDER	MAP DATE	MAP TIME	FIELD COUNT	RECORD COUNT
ABCD01M	1	G	08/12/99	130517	0	0

Figure 4-8. Sample CREPORT 032

REPORT NO. 33 CREPORT 033		CA-IDMS/DC MAPPING REPORT REL 15.0 LISTING OF MAPS				09/08/99 PAGE 1
MAP NAME	MAP VERSION	MAP BUILDER	PANEL NAME	PANEL VERSION	MAP DATE	MAP TIME
ABCD01M	1	G	ABCD01M-OLMPANEL	1	08/12/99	130517
ABIF01M	1	G	ABIF01M-OLMPANEL	1	02/26/99	150630
ADDS01M	1	G	ADDS01M-OLMPANEL	1	05/04/99	104421
ADMI01M	1	G	ADMI01M-OLMPANEL	1	03/02/99	112042
ADMS01M	1	G	ADMS01M-OLMPANEL	1	08/13/99	155714

Figure 4-9. Sample CREPORT 033

REPORT NO. 34 CREPORT 034		CA-IDMS/DC MAPPING REPORT REL 15.0 LISTING OF MAPS BY RECORD NAME RECORD NAME: ACEXE00M-MAP-RECORD RECORD VERSION: 1				09/08/99 PAGE 1
MAP NAME	MAP VERSION	PANEL NAME	PANEL VERSION			
AD21M007	1	AD21M007-OLMPANEL	1			
AD21M008	1	AD21M008-OLMPANEL	1			
AD21M008	1	AD21M008-OLMPANEL	1			
AD21M010	1	AD21M010-OLMPANEL	1			

Figure 4-10. Sample CREPORT 034

REPORT NO. 35 CREPORT 035		CA-IDMS/DC MAPPING REPORT REL 15.0 LISTING OF MAPS BY ELEMENT NAME ELEMENT NAME: ADMIT-DATE-0430 OF RECORD NAME: HOSPITAL-CLAIM RECORD VERSION: 1				09/08/99 PAGE 1
MAP NAME	MAP VERSION	PANEL NAME	PANEL VERSION			
A310M4	1	A310M4-OLMPANEL	1			

Figure 4-11. Sample CREPORT 035

## 4.10.2 Field descriptions

Because many of the field names on the map reports are the same from report to report, a description of all the fields is presented once in alphabetical order.

### **ALARM**

Indicates whether the terminal's audible alarm will sound automatically when the map is written out to the screen (CREPORTs 030 and 031 only).

### **BUILDER**

Identifies the builder code for the record that defines the map (CREPORT 032 only). For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

### **CURSOR**

Identifies the name of the panel field in which the cursor appears after a mapout operation (CREPORTs 030 and 031 only).

### **ELEMENT NAME**

Identifies the name of a data field (record element) whose map associations are being described (CREPORT 035 only).

### **FIELD COUNT**

Indicates the number of nonliteral fields in the map (CREPORTs 030, 031, and 032 only).

### **FIELD NAME**

Identifies the panel fields that appear in the map (CREPORT 031 only).

### **INDEX**

Indicates the order in which the record or record fields are used by the map (CREPORTs 030 and 031 only).

### **MAP BUILDER**

Identifies the builder code for the record that defines the map occurrence (CREPORT 033 only). For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

### **MAP DATE**

Identifies the date the map occurrence was defined to the dictionary (CREPORTs 030, 031, 032, and 033 only).

### **MAP NAME and MAP VERSION**

Identify the map occurrence being described.

### **MAP TIME**

Identifies the time the map was last compiled with critical changes (CREPORTs 030, 031, 032, and 033 only).

### **PANEL NAME and PANEL VERSION**

Identify the panel associated with the map being described (CREPORTs 032, 033, 034, and 035 only).

**PANEL BUILDER**

Identifies the builder code for the record that defines the panel occurrence (CREPORT 032 only). For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

**RECORD COUNT**

Indicates the number of records used by the map.

**RECORD NAME**

Identifies the record used by the map (CREPORTs 030, 031, and 032 only).

**RESET**

Indicates whether all modified data tags are reset when the map is mapped out (YES) or remain unchanged (NO) (CREPORTs 030 and 032 only).

**UNLOCK**

Indicates whether the keyboard will be locked (NO) or unlocked (YES) when the map is mapped out (CREPORTs 030 and 032 only).

## 4.11 Nodes and Resource Table Reports (CREPORTS 043 and 044)

### 4.11.1 Contents

The Nodes and Resource Table reports provide information on all the resources that have been defined to the systems, including the location where the resource resides and the type of communication method used to access these nodes. The table below summarizes each report:

CREPORT 043	Listing of Nodes	Lists all the nodes sorted on access type
CREPORT 044	Listing of Defined Resources	Lists all the resources sorted on database or destination name

REPORT NO. 43 CREPORT 043		CA-IDMS/DC NODE NAME REPORT LISTING OF NODES				REL 15.0	09/08/99 PAGE 1	
SYSTEM VERSION NUMBER	NODE NAME	ACCESS TYPE	DEFAULT NODE	CV NUMBER	SVC NUMBER	BUILDER CODE	ACTION CODE	
71	A06IVT12	CCI				R		
71	SYSTEM22	CCI				R		
71	SYSTEM92	CCI				R		
71	DBDCGR	GROUP	NULL			R		
71	DBGNOD1	GROUP	NULL			R		
71	DBGNOD2	GROUP	LOCAL			R		
71	DBGNOD3	GROUP	SYSTEM72			R		
71	IDMSGR	GROUP	NULL			R		
71	SYSTEM71	LOCAL				R		
71	CVNOD1	SVC		101	173	R		
71	CVNOD2	SVC		102	102	R		
71	SYSTEM72	VTAM				R		
71	SYSTEM73	VTAM				R		
71	SYSTEM74	VTAM				R		
71	TECHDC99	VTAM				R		

Figure 4-12. Sample CREPORT 043

REPORT NO. 44 CREPORT 044		CA-IDMS/DC RESOURCE TABLE REPORT LISTING OF DEFINED RESOURCES			REL 15.0	09/08/99 PAGE 1	
SYSTEM VERSION NUMBER		DATABASE NAME	DESTINATION	NODE NAME	BUILDER CODE	ACTION CODE	
71		DBNAM1		LOCAL	R		
71		DBNAM2		DBGNOD1	R		
71			DBGNOD2	LOCAL	R		

Figure 4-13. Sample CREPORT 044

## 4.11.2 Field descriptions

A description of the fields in the sample reports follows:

**SYSTEM VERSION NUMBER**

Identifies the version number of the system associated with the nodes or resources.

**NODE NAME**

Identifies the name of the node.

**ACCESS TYPE**

Specifies the communication method used to access the corresponding node.

**DEFAULT NODE**

For access type GROUP only. Identifies the default node to use if access to the requested group fails.

**CV NUMBER**

For access type SVC only. Identifies the number of the central version.

**SVC NUMBER**

For access type SVC only. Identifies the number of the SVC through which the system will send packets to the corresponding node.

**BUILDER CODE**

Identifies the builder code for the record that defines the node or resource.

For a description of building codes, see Table 4-1 at the end of this chapter.

**ACTION CODE**

Identifies whether the node or resource occurrence has been updated (U) or deleted (D). A blank indicates that it has not been changed since the last time the system was generated.

**DATABASE NAME**

Identifies the name of the database that is included in the resource name table.

**DESTINATION**

Identifies a nodename to which requests for data can be sent.

## 4.12 Module Text to Card Utility (CREPORT 051)

**Purpose:** The Module Text to Card utility lets you punch the module source code of a specified module to cards. The CREPORT and KEY parameters used to punch module CREPORT 011 to card are:

```
CREPORT=051
KEY MOD-NAME-067 'CREPORT 011'
```

**How to run CREPORT 051:** You must run CREPORT 051 alone. To run CREPORT 051, include the following specification in the JCL:

■ **For OS/390 systems:**

```
//SYSPCH DD SYSOUT=B,DCB=BLKSIZE=80
```

■ **For VSE/ESA systems:**

```
// ASSGN SYSPCH,X'ccc'
```

---

ccc	device assignment (channel and unit) for punched output
-----	---------------------------------------------------------

---

■ **For VM/ESA systems:**

```
FILEDEF SYSPCH DISK syspch output a
```

---

syspch output a	file identifier of the card-image output file
-----------------	-----------------------------------------------

---

■ **For BS2000/OSD systems:**

```
/CREATE-FILE F-NAME=temp.punchcard,(SPACE=(primary,secondary)), -
/ SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYSPCH,F-NAME=temp.punchcard
```

---

temp.punchcard	file name of punched card output file
----------------	---------------------------------------

---

primary	primary space allocation
---------	--------------------------

---

secondary	secondary space allocation
-----------	----------------------------

---

After the CA-CULPRIT job runs, you must use the BS2000/OSD PUNCH user command to punch the temp.punchcard file.

**How to request card-image listing:** No *printed* output is produced by this report other than an accounting of the number of records written. To request a card-image listing of the module instead of punched cards, include the following JCL specification:

■ **For OS/390 systems:**

```
//SYSPCH DD SYSOUT=A
```

■ **For VSE/ESA systems:**

```
// ASSGN SYSPCH,X'ppp'
```

---

ppp	printer device assignment
-----	---------------------------

---

■ **For VM/ESA systems:**

```
FILEDEF SYSPCH PRINTER
```

■ **For BS2000/OSD systems:**

```
/CREATE-FILE F-NAME=temp.lstf,(SPACE=(primary,secondary)), -
/ SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYSPCH,F-NAME=temp.lstf
```

---

temp.lstf	file name of printed output file
primary	primary space allocation
secondary	secondary space allocation

---

After the CA-CULPRIT job runs, you must use the BS2000/OSD PRINT user command to print the temp.lstf file.



## 4.13 Module Text to File Utility (CREPORT 052)

**Purpose:** The module text to output file utility (CREPORT 052) lets you output module source code to a disk file. The CREPORT and KEY parameters used to output module CREPORT 011 to file are shown below:

```
CREPORT=052
KEY MOD-NAME-067 'CREPORT 011'
```

**How to run CREPORT 052:** CREPORT 052 must be run alone. To run CREPORT 052, add the following specification to the JCL:

■ **For OS/390 files:**

```
//SYS020 DD DSN=user.textfile,DISP=(NEW,CATLG),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=320),
//          UNIT=disk,VOL=SER=nnnnnn
```

user.textfile	data set name of the output file
disk	symbolic device name of disk
nnnnnn	volume serial number of disk

■ **For VSE/ESA tape files:**

```
// TLBL  SYS020,'user.text'
// ASSGN SYS020,TAPE,VOL=nnnnnn
```

user.text	file-id of tape file
nnnnnn	tape volume serial number

■ **For VSE/ESA disk files:**

```
// DLBL  SYS020,'user.text'
// EXTENT SYS020,nnnnnn
// ASSGN  SYS020,DISK,VOL=nnnnnn,SHR
```

user.text	file-id of disk file
nnnnnn	volume serial number of the disk file

■ **For VM/ESA files:**

```
FILEDEF SYS020 DISK nonprint file a (RECFM FB LRECL 80 BLKSIZE 320
```

nonprint file a	filename, filetype, and filemode of the nonprint/nonpunch output file
-----------------	-----------------------------------------------------------------------

■ **For BS2000/OSD files:**

```
/CREATE-FILE F-NAME=user.nonprint, (SPACE=(primary,secondary)),      -  
/  SUPPRESS-ERR=*FILE-EXIST  
/ADD-FILE-LINK L-NAME=SYS020, F-NAME=user.nonprint
```

---

user.nonprint	file name for nonprint/nonpunch output file
primary	primary space allocation
secondary	secondary space allocation

---

## 4.14 Network Description Reports (CREPORTs 001-003, 014-018)

### 4.14.1 Contents

Network reports (CREPORTs 001, 002, 003, 014, 015, 016, 017, and 018) describe the lines, physical terminals, and logical terminals associated with a DC/UCF system. CREPORTs 001, 002, and 003 describe the networks that have been defined for executable systems; CREPORTs 014 through 018 describe networks for systems that have been defined but not generated.

#### Summary of network description reports

CREPORT Module	CREPORT Name
001	Network Description by Line (Object Report)
002	Network Description by Physical Terminal (Object Report)
003	Network Description by Logical Terminal (Object Report)
014	Network Description by Line (Source Report)
015	Network Description by Physical Terminal (Source Report)
016	Physical Terminals within Line (Source Report)
017	Network Description by Logical Terminal (Source Report)
018	Logical Terminal by Physical Terminal (Source Report)

### 4.14.2 Sample reports

The following figures show sample output for CREPORTs 001, 002, 016, and 018. Sample output for CREPORTs 014, 015, and 017 is not shown; the fields in these three reports are the same as those in CREPORTs 001, 002, and 003, respectively.

#### 4.14 Network Description Reports (CREPORTs 001-003, 014-018)

REPORT NO. 01			CA-IDMS/DC SYSTEM GENERATION REPORT							REL 15.0		09/08/99 PAGE 1			
CREPORT 001			SYSTEM NAME: DCSYSTEM												
			SYSTEM VERSION: 99												
			NETWORK DESCRIPTION BY LINE												
NAME			BUILDER ACTION VERSION												
LINE	CONSOLE	R	1	TYPE	BUFFSIZE	-----DDNAMES-----			DISABLED						
				CONSOLE	0				NO						
PTRM	OPERATOR	R	1	TYPE	MAX-ERR	LINE-LEN	PAGE-LEN	MODEL	DISABLED	ASCII	LNDEL	CHRDDEL	CANCEL	PRINTER	CLASS
				OPER	3	0	0		NO	NO	00	00	00		1
LTERM	CONSOLE	R	1	PRIORITY	CASE	DISABLED	PRINTER	TASK CODE							
				240	UPPER	NO	NO								

Figure 4-14. Sample CREPORT 001

REPORT NO. 02 CREPORT 002			CA-IDMS/DC SYSTEM GENERATION REPORT REL 15.0 SYSTEM NAME: DCSYSTEM SYSTEM VERSION: 99 NETWORK DESCRIPTION BY PHYSICAL TERMINAL										09/08/99 PAGE 1					
PTRM NAME	LINE NAME	LTERM NAME	BUILDER	ACTION	VERS	LINE TYPE	DEV TYPE	MAX ERR	LINE LGTH	PAGE LGTH	MODEL	DISABL	ASCII	LNDEL	CHRDDEL	CANCEL	PRT CLS	DEST
OPERATOR	CONSOLE	CONSOLE	R		1	CONSOLE	OPER	3	0	0		NO	NO	00	00	00	1	
UCFPTB1	UCFLINE	UCFLTB1	R		1	UCFLINE	2F	3	0	0		NO	NO	00	00	00	1	
UCFPTB2	UCFLINE	UCFLTB2	R		1	UCFLINE	2F	3	0	0		NO	NO	00	00	00	1	
UCFPTB3	UCFLINE	UCFLTB3	R		1	UCFLINE	2F	3	0	0		NO	NO	00	00	00	1	
UCFPTB4	UCFLINE	UCFLTB4	R		1	UCFLINE	2F	3	0	0		NO	NO	00	00	00	1	

Figure 4-15. Sample CREPORT 002

REPORT NO. 16 CREPORT 016		CA-IDMS/DC SOURCE REPORT REL 15.0 SYSTEM NAME: DCSYSTEM SYSTEM VERSION: 99 PHYSICAL TERMINALS WITHIN LINE LINE NAME: UCFLINE										09/08/99 PAGE 2		
PTERM NAME	BUILDER	ACTION	VERSION	TYPE	MAX-ERR	LINE-LEN	PAGE-LEN	MODEL	DISABLED	ASCII	LNDEL	CHRDEL	CANCEL	P-CLASS
UCFPTB1	G		1	2F	3	0	0	00	NO	NO	00	00	00	1
UCFPTB2	G		1	2F	3	0	0	00	NO	NO	00	00	00	1
UCFPTB3	G		1	2F	3	0	0	00	NO	NO	00	00	00	1
UCFPTB4	G		1	2F	3	0	0	00	NO	NO	00	00	00	1
UCFPTB5	G		1	2F	3	0	0	00	NO	NO	00	00	00	1

Figure 4-16. Sample CREPORT 016

REPORT NO. 18 CREPORT 018			CA-IDMS/DC SOURCE REPORT REL 15.0 SYSTEM NAME: DCSYSTEM SYSTEM VERSION: 99 LOGICAL TERMINAL BY PHYSICAL TERMINAL					09/08/99 PAGE 1	
PTERM NAME	LTERM NAME	BUILDER	ACTION	VERSION	PRIORITY	CASE	DISABLED	PRINTER	TASK CODE
OPERATOR	CONSOLE	G		1	240	UPPER	NO	NO	
UCFPTB1	UCFLTB1	G		1		UPPER	NO	NO	
UCFPTB2	UCFLTB2	G		1		UPPER	NO	NO	
UCFPTB3	UCFLTB3	G		1		UPPER	NO	NO	
UCFPTB4	UCFLTB4	G		1		UPPER	NO	NO	

Figure 4-17. Sample CREPORT 018

### 4.14.3 Field descriptions

Because many of the field names in the network reports are the same from report to report, all the fields in the sample reports are described once, in alphabetical order:

#### **ACTION**

Indicates whether the entity occurrence being described has been updated (U) or deleted (D) or remain unchanged (blank) since the most recent generate.

#### **APPLICATION ID**

Defines the VTAM application id used by the system at run time to sign the line on to VTAM (CREPORTs 001 and 014 only).

#### **APPLICATION PW**

Identifies the optional password used by the system to sign on to VTAM (CREPORTs 001 and 014 only).

#### **ASCII**

Indicates whether the physical terminal supports the ASCII character set (CREPORTs 001, 002, 014, 015, and 016 only).

#### **BUFFSIZE**

Identifies the line I/O page buffer size, in bytes, for the line (CREPORTs 001 and 014 only).

#### **BUILDER**

Identifies the builder code for the record that defines the entity occurrence. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

#### **CANCEL**

Identifies the control character used as the attention key by the physical terminal device type (CREPORTs 001, 002, 014, 015, and 016 only).

#### **CASE**

Identifies the character set the logical terminal uses on input (CREPORTs 001, 003, 014, 017, and 018 only).

**CHRDEL**

Identifies the control character that is used to delete characters (CREPORTs 001, 002, 014, 015, and 016 only).

**DDNAMES**

Identifies the ddname and/or file id for the line, as specified in the system startup JCL (CREPORTs 001 and 014 only).

**DEST**

Specifies the line destination.

**DEV TYPE or TYPE**

(Lines only) Identifies the generic linetype for the terminals that are included in the line group (CREPORTs 001, 002, 014, 015, and 016 only).

**LINE-LEN/LINE LGTH**

Identifies the maximum line length in characters for the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

**DISABLED/DISABL**

Indicates whether the line group, physical terminal, or logical terminal is disabled when the system is started.

**LNDEL**

Identifies the control character (flush character) that is used to delete lines (CREPORTs 001, 002, 014, 015, and 016 only).

**MAX-ERR**

Indicates the number of retries allowed after a terminal I/O error before the teleprocessing monitor disables the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

**MODEL**

Identifies the model number of the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

**NAME**

Identifies the entity being described (can be preceded by SYSTEM, LINE, LTERM, or PTERM).

**PAGE-LEN**

Identifies the maximum page size, in text lines, for the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

**PRINTER/PRINT**

(IDMS-DC only) Indicates whether the logical terminal is a 3280-type printer (CREPORTs 001, 003, 014, 017, and 018 only).

**PRINTER CLASS(ES)/PRT CLS/P-CLASS**

(IDMS-DC only) Defines the default printer class for the physical terminal.

**PRIORITY/PRIOR**

Identifies the dispatching priority for requests to IDMS-DC from the logical terminal (CREPORTs 001, 003, 014, 017, and 018 only).

**REQUEST PARAM LIST**

Indicates the number of write-only input/output requests that can be handled at one time by the local VTAM line group (CREPORTs 001 and 014 only).

**TASK CODE**

Identifies the task that is automatically executed whenever the ENTER NEXT TASK CODE would normally be displayed and when a DC RETURN request that specifies no task code is issued (CREPORTs 001, 003, 014, 017, and 018 only).

**TYPE**

(Physical terminals only) Identifies the device type of the physical terminal. See DEV TYPE above for a description of line type.

**VERSION**

Identifies the version number of the entity occurrence being described.

## 4.15 CA-OLQ Reports (CREPORTs 041 and 046)

### 4.15.1 Contents

CA-OLQ reports provide information about CA-OLQ run-time parameters that have been defined for each system. CA-OLQ parameters are represented in the dictionary by the CVGDEFS-142 record, which is a logical extension of the SYS-041 record.

CREPORT 041 describes CA-OLQ parameters for executable systems; CREPORT 046 describes CA-OLQ parameters for systems that have been defined but not generated. The fields and format of the two reports are the same.

REPORT NO. 41 CREPORT 041		CA-IDMS/DC OLQ REPORT REL 15.0 LISTING OF OLQ PARAMETERS OBJECT REPORT										09/08/99 PAGE 1	
SYSTEM VERSION	OLQ TRANS ID												
99	OLQ	PFKEY MOD NAME	BUILDER CODE R	ACTION CODE	PRINT LINE SIZE 80	PRINT PAGE SIZE 60	INTERUPT COUNT 100	INT STORAGE PAGE SIZE 1,920	REPORT FILE PAGE SIZE 4,000	INPUT LINE SIZE 4	REPORT RETENTION 1	MAXIMUM RETENTION 5	
		REPORT DICT	SQL ACCESS OLQSQL	BATCH CLASS 0	CONTINUATION CHAR -	SEPARATOR CHAR !	COMMENT CHAR ;	MENU MODE ALLOWED	MAX INTERRUPT	MAX REP PAGES 0	MAX REP COUNT 5	MAX SORT STORAGE 40	
105	OLQ	PFKEY MOD NAME	BUILDER CODE R	ACTION CODE	PRINT LINE SIZE 80	PRINT PAGE SIZE 60	INTERUPT COUNT 100	INT STORAGE PAGE SIZE 1,920	REPORT FILE PAGE SIZE 4,000	INPUT LINE SIZE 4	REPORT RETENTION 1	MAXIMUM RETENTION 5	
		REPORT DICT	SQL ACCESS OLQSQL	BATCH CLASS 0	CONTINUATION CHAR -	SEPARATOR CHAR !	COMMENT CHAR ;	MENU MODE ALLOWED	MAX INTERRUPT	MAX REP PAGES 0	MAX REP COUNT 5	MAX SORT STORAGE 40	

Figure 4-18. Sample CREPORT 041

### 4.15.2 Field descriptions

A description of the fields in the sample report follows:

#### SYSTEM VERSION

Identifies the system whose OLQ run-time definitions are being described.

#### OLQ TRANS ID

Identifies the task code used to invoke the CA-OLQ run-time system.

#### PFKEY MOD NAME

Identifies the module that contains the default control key assignments for CA-OLQ.

#### BUILDER CODE

Identifies the builder code for the record that defines the CA-OLQ parameters.  
For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.



**ACTION CODE**

Indicates whether the record that defines the CA-OLQ parameters has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate.

**PRINT LINE SIZE**

Identifies the line length, in characters, for CA-OLQ report output on TTY-type terminals.

**PRINT PAGE SIZE**

Identifies the page length, in lines, for CA-OLQ report output on TTY-type terminals.

**INTERRUPT COUNT**

Indicates the maximum number of records that will be read by CA-OLQ before requesting further terminal input.

**INT STORAGE PAGE SIZE**

Indicates the size, in bytes, of the CA-OLQ internal storage pages used to store control data across a pseudo-converse.

**REPORT FILE PAGE SIZE**

Indicates the size, in bytes, for CA-OLQ report file pages written to the DDLDCRUN area.

**INPUT LINE SIZE**

Identifies the number of lines on the screen that are available for input.

**REPORT RETENTION**

Identifies the default report retention time, in days.

**MAXIMUM RETENTION**

Identifies the maximum report retention time, in days.

**REPORT DICT**

Identifies the name of the dictionary in which catalog information about CA-OLQ saved reports is stored.

**SQL ACCESS**

Indicates how SQL statements used to access a CA-IDMS/DB database will be processed. OLQSQL indicates CA-OLQ will process the statements; IDMSSQL indicates CA-IDMS/DB will process the statements.

**BATCH CLASS**

(OS/390 only) Identifies the print class used by CA-OLQ when submitting batch jobs.

**CONTINUATION CHAR**

Identifies the continuation character for CA-OLQ.

**SEPARATOR CHAR**

Identifies the separation character for CA-OLQ.

**COMMENT CHAR**

Identifies the comment character for CA-OLQ.

**MENU MODE**

Indicates whether menu mode is allowed.

**MAX INTERRUPT**

Indicates the maximum interrupt count that a user can specify at run time.

**MAX REP PAGES**

Indicates the maximum report size, in pages.

**MAX REP COUNT**

Indicates the maximum number of reports each user can save in the DDLDCRUN area.

**MAX SORT STORAGE**

Indicates the maximum amount of storage, in K bytes, that CA-OLQ can use for sort operations.

## 4.16 Program Description Reports (CREPORTs 004 and 019)

### 4.16.1 Contents

Program description reports provide information about programs that have been included in a DC/UCF system. System-supplied programs, subschemas, database procedures, maps, edit and code tables, CA-ADS dialogs, and user programs written in COBOL, PL/I, and Assembler are represented in the dictionary as program occurrences.

CREPORT 004 describes programs associated with executable systems; these programs are represented in the dictionary by the PROGLST-049 record. CREPORT 019 describes programs associated with systems that have been defined but not generated; these programs are represented by the PROG-051 record. The fields and format of the two reports are the same.

REPORT NO. 04		CA-IDMS/DC SYSTEM GENERATION REPORT					REL 15.0		09/08/99 PAGE			1
CREPORT 004		SYSTEM NAME: DCSYSTEM										
		SYSTEM VERSION: 99										
		PROGRAM DESCRIPTION										
PROGRAM	TYPE	BUILDER ACTION VERSION										
\$ACF@GEN	TABLE	R	U	1	ISA-SIZE	ERROR	DUMP			SAVE		
					0	THRESHOLD	THRESHOLD	LOAD FROM	LANGUAGE	AREA	RESIDENT	
					CONCURRENT	5	0	LOADLIB	ASSEMBLER	NO	NO	
					YES	REUSABLE	REENTRANT	OVERLAYABLE	DISABLED	PROTECT	NEW COPY	
					MAINLINE	YES	REENTRANT	NO	NO	YES	NO	
					DIALOG	EXCLUDE	DIALOG					
\$ACF@TAT	TABLE	R	U	1	ISA-SIZE	ERROR	DUMP			SAVE		
					0	THRESHOLD	THRESHOLD	LOAD FROM	LANGUAGE	AREA	RESIDENT	
					CONCURRENT	5	0	LOADLIB	ASSEMBLER	NO	NO	
					YES	REUSABLE	REENTRANT	OVERLAYABLE	DISABLED	PROTECT	NEW COPY	
					MAINLINE	YES	REENTRANT	NO	NO	YES	NO	
					DIALOG	EXCLUDE	DIALOG					
					ISA-SIZE	ERROR	DUMP			SAVE		
					0	THRESHOLD	THRESHOLD	LOAD FROM	LANGUAGE	AREA	RESIDENT	
					CONCURRENT	5	0	LOADLIB	ASSEMBLER	NO	NO	
					YES	REUSABLE	REENTRANT	OVERLAYABLE	DISABLED	PROTECT	NEW COPY	
					MAINLINE	YES	REENTRANT	NO	NO	YES	NO	
					DIALOG	EXCLUDE	DIALOG					
					ISA-SIZE	ERROR	DUMP			SAVE		
					0	THRESHOLD	THRESHOLD	LOAD FROM	LANGUAGE	AREA	RESIDENT	
					CONCURRENT	5	0	LOADLIB	ASSEMBLER	NO	NO	
					YES	REUSABLE	REENTRANT	OVERLAYABLE	DISABLED	PROTECT	NEW COPY	
					MAINLINE	YES	REENTRANT	NO	NO	YES	NO	
					DIALOG	EXCLUDE	DIALOG					

Figure 4-19. Sample CREPORT 004

### 4.16.2 Field descriptions

A description of the fields in the sample report follows:

#### SYSTEM NAME and VERSION

Identify the name and version of the system associated with the program occurrences being described.

#### PROGRAM

Identifies the program being described.

#### TYPE

Identifies the program occurrence as a PROGRAM, MAP, SUBSCHEMA, TABLE, DIALOG, ACCESS MODULE, or MAP HELP.

**BUILDER**

Identifies the builder code for the record that defines the program occurrence. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

**ACTION**

Indicates whether the program occurrence has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate.

**VERSION**

Identifies the version number associated with the program occurrence.

**ISA-SIZE**

(IDMS-DC only) Indicates the amount of storage, in bytes, that is allocated for the program's initial storage area (ISA) (applies to Assembler and PL/I programs only).

**ERROR THRESHOLD**

(IDMS-DC only) Indicates the number of program check errors that can occur before the program is disabled by the system.

**DUMP THRESHOLD**

(IDMS-DC only) Indicates the maximum number of dumps to be taken for program check errors that occur in the program.

**LOAD FROM**

Indicates whether the program resides in a load library (LOADLIB) or in the load area of the dictionary (DICTIONARY).

**LANGUAGE**

Identifies the source language of the program.

**SAVE AREA**

Indicates whether a save area is acquired automatically before each execution of the program.

**RESIDENT**

Indicates whether the program is made resident when the DC/UCF system is started (YES) or whether the program is a nonresident program that is loaded into the storage pool on request (NO).

**CONCURRENT**

Indicates whether the program can process more than one request concurrently.

**REUSABLE**

Indicates whether the program is reusable.

**REENTRANT**

Indicates whether the program is fully reentrant, quasi-reentrant, or nonreentrant.

**OVERLAYABLE**

(IDMS-DC only) Indicates whether the program can be overlaid in the program pool.

**DISABLED**

Indicates whether the program is disabled when the DC/UCF system is started.

**PROTECT**

Indicates whether the DC/UCF storage protection feature is in effect for the program.

**NEW COPY**

Indicates whether the new copy facility is enabled.

**MAINLINE DIALOG**

Indicates whether (YES) or not (NO) the program is a CA-ADS mainline dialog.

**EXCLUDE FROM MENU**

For a CA-ADS dialog, indicates whether (YES) or not (NO) the dialog will appear on the CA-ADS menu screen.

**DIALOG STATS ON**

For a CA-ADS dialog, indicates whether (YES) or not (NO) statistics are collected.

**DYNAMIC**

Indicates whether (YES) or not (NO) users are allowed to define additional versions of the program at run time either by means of the DCMT VARY DYNAMIC PROGRAM command or, if the program is eligible for automatic definition, through definition of null PDEs on the SYSTEM system generation statement.

**MPMODE**

Identifies the multiprocessing mode (MPMODE) for the program. SYSTEM directs DC/UCF to assign a mode to the program at execution time. ANY specifies an MPMODE of ANY.

## 4.17 Queue Description Reports (CREPORTs 006, 022, and 023)

### 4.17.1 Contents

Queue description reports provide information about queue occurrences associated with a DC/UCF system. CREPORT 006 describes queues associated with executable systems; these queues are represented in the dictionary by the QUEUELST-029 record. CREPORTs 022 and 023 describe queues associated with systems that have been defined but not generated; these queues are represented by the QUEUE-030 record. The fields and format of the three reports are almost identical.

REPORT NO. 06 CREPORT 006		CA-IDMS/DC SYSTEM GENERATION REPORT					REL 15.0		09/08/99 PAGE 1	
		SYSTEM NAME: DCSYSTEM								
		SYSTEM VERSION: 99								
		QUEUE DESCRIPTION								
QUEUE	PROGRAM	TASK	BUILDER	ACTION	VERSION	THRESHOLD VALUE	UPPER LIMIT	DISABLED	QUEUE RETENTION	
OLQNOTE	OLQNOTE	OLQNOTE	R	U	1	1	0	NO	1	

Figure 4-20. Sample CREPORT 006

### 4.17.2 Field descriptions

A description of the fields in the sample report follows:

#### **SYSTEM NAME and VERSION**

Identifies the name and version number of the system associated with the queues being described.

#### **QUEUE**

Identifies the queue being described.

#### **PROGRAM**

Identifies the name of the initial program invoked by the task associated with the queue.

#### **TASK**

Identifies the task code for the task invoked when the number of entries in the queue reaches the limit defined with the THRESHOLD VALUE parameter (below).

#### **BUILDER**

Identifies the builder code for the record that defines the queue occurrence. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

**ACTION**

Indicates whether the queue occurrence has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate.

**VERSION**

Indicates the version number of the queue being described.

**THRESHOLD VALUE**

Indicates the number of entries that must be in the queue before the system invokes the task associated with the queue.

**UPPER LIMIT**

Indicates the maximum number of records that can be directed to the queue.

**DISABLED**

Indicates whether the queue is disabled at system startup.

**QUEUE RETENTION**

Indicates the queue retention period in days.

## 4.18 Symbol Table Report (CREPORT 053)

### 4.18.1 Contents

The symbol table report lists information about symbol table load modules stored in the system dictionary DDLDCLOD area.

REPORT NO. 53 CREPORT 053		CA-IDMS/DC LOAD AREA REPORT SYMBOL TABLES				REL 15.0	09/08/99 PAGE 1
MODULE NAME	VERSION	NUM RLD ENTRIES	ENTRY PT ADDRESS	MODULE LENGTH	COMPILE DATE	COMPILE TIME	
ADMI01D	1	7	0	5,036	08/19/99	112126	
ADOLFIX	1	8	0	6,008	05/06/99	153217	
ADOL01D	1	7	0	7,976	08/16/99	144254	
ADRP01D	1	14	0	2,768	07/15/99	101632	
ANSI01D	1	7	0	6,144	03/11/99	163656	
BIFX01D3	1	21	0	3,408	02/25/99	095854	
BIFX02D	1	21	0	4,600	02/26/99	140559	
CLSTAX1	1	7	0	3,424	03/19/99	143936	
CLSTOX1	1	7	0	3,424	03/22/99	111623	
CLSTRHDX	1	7	0	3,424	03/22/99	150911	
CLSTRHDY	1	7	0	3,424	03/22/99	152112	

Figure 4-21. Sample CREPORT 053

### 4.18.2 Field descriptions

A description of the fields in the sample report follows:

**MODULE NAME**

Name of the symbol table load module.

**VERSION**

Version number of the symbol table load module.

**NUM RLD ENTRIES**

Number of entries in the relocation dictionary (RLD) for the symbol table load module.

**ENTRY PT ADDRESS**

Entry point address of the symbol table load module.

**MODULE LENGTH**

Length, in bytes, of the object text for the symbol table load module.

**COMPILE DATE**

Date the symbol table load module was compiled (*mm/dd/yy*).

**COMPILE TIME**

Time the symbol table load module was compiled (*hhmmss*).



## 4.19 System Options Reports (CREPORTs 011 and 025)

### 4.19.1 Contents

System options reports provide detailed information about DC/UCF systems that have been defined to the dictionary. CREPORT 011 describes the options defined for executable systems; these systems are represented in the dictionary by the SYSMO-170 record. CREPORT 025 describes options defined for systems that have been defined but not generated; these systems are represented by the SYS-041 record. The fields and format of the two reports are the same.

REPORT NO. 11 CREPORT 011		CA-IDMS/DC SYSTEM GENERATION REPORT					REL 15.0	09/08/99	PAGE	1
DC SYSTEM OPTIONS - SYSGEN										
SYSTEM NAME DCSYSTEM		VERSION 99	BUILDER R	GENERATION ID	SYSTEM ID SYST0099	OPERATING SYSTEM OS/390		DESCRIPTION		
DATE CREATED: 08/16/99		PREPARED BY:		DATE LAST USED:		REVISED BY:				
DC PARAMETERS:										
STORAGE ALLOCATION:		CWA SIZE 0	PROGRAM POOL(K) 512	STORAGE CUSHION(K) 0	STORAGE POOL(K) 751	REENTRANT POOL(K) 0	XA PROGRAM POOL(K) 0	XA STORAGE POOL(K) 0	XA REENTRANT POOL(K) 0	RELOCATABLE THRESHOLD(%) 0
RUNTIME ENVIRONMENT:		MAXIMUM TASKS 21	MSGDICT 1	SCRATCH/QUEUE 1	SIGNON/DEST 1	SYSTEM RUN UNITS 1	SECURITY 1	SYSTEM/DEST 1	STORAGE PROTECTION NO	ALTERNATE PROT KEY 9
		AUTOMATIC NEW COPY NO	SUBSCHEMAS NO	UNDEFINED MAPS NO	LOADABLE TABLES NO	DIAGLOGS NO	ACCESS MODULES NO	PRINTED REP RET 7	PRINTER CHECKPOINT OFF	RESOURCE INTERVAL OFF
		OLQ YES	ADSO YES	PF KEYS YES	OLM KEYS YES	PAGE REL NO	LOADLIST	XA SCRATCH NO	TIMEOUT - - RHDCBYE 1	
INTERNAL LIMITS: OFF		STG LIMIT 0	LOCK LIMIT 0	CALL LIMIT 0	DBIO LIMIT 0	EXTERNAL LIMITS: OFF	STG LIMIT 0	LOCK LIMIT 0	CALL LIMIT 0	DBIO LIMIT 0
OPERATING SYSTEM INTERFACE		LOG TO DATABASE NO	- - FILE 1 CDMSLOGA	LOGGING COUNT 1 0	INFORMATION - - FILE 2 COUNT 2 1	LOG DEVICE DOS ONLY	SVC NUMBER			
		3270 PRINT KEY OFF	WTO DESCRIPTION CODES NONE				WTO ROUTE CODES NONE			
ERROR HANDLING:		INACTIVE INTERVAL NONE	RUNAWAY INTERVAL 10	SYSTEM DUMP NO	SYSTEM TRACE ON	TRACE ENTRIES 250	USER TRACE OFF	USERTRACE ENTRIES 0	ABRU SNAP NO	DEADLOCK INTERVAL 5
SYSTEM INTERNALS:		ABEND STORAGE(WDS) 200	DPE COUNT 431	ECB LIST(WDS) 42	RCE COUNT 525	RLE COUNT 840	STACKSIZE (WDS) 1200	PRIMARY NULL PDES 0	SECONDARY NULL PDES 0	
STATISTICS OPTIONS:		TIME INTERVAL 0	BY TASK NO		BY TRAN NO	BY LINE NO	BY USER NO	COLLECT		
DB PARAMETERS:		JOURNAL RETRIEVAL JOUR	MAXIMUM SYSLOCKS 10	AREA THRESHOLD OFF	AREA RETRY FOREVER	RETRIEVAL LOCKING NOLOCK	UPDATE LOCKING NOLOCK	SCRATCH/QUEUE JOURNAL BEFORE ONLY NO	JOURNAL FRAGMENT OFF	JOURNAL TRANSACTION 0
CV PARAMETERS:		CHKUSER TASKS 0	EXTERNAL WAIT 600	INTERNAL WAIT 1800	MAXIMUM ERUS 6	RUPRTY 100				
SYSCTL PARAMETERS:		DD NAME SYSCTL	DB NAME DEFAULT	NODE NAME DEFAULT						

Figure 4-22. Sample CREPORT 011

## 4.19.2 Field descriptions

A description of the fields in the sample report follows:

### **SYSTEM NAME and VERSION**

Identifies the DC/UCF system being described.

### **BUILDER**

Identifies the builder code for the record that defines the system occurrence. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

### **GENERATION ID**

Identifies the unique identifier for the system, as defined in the system options table at startup.

### **SYSTEM ID**

Indicates the name (nodename) by which the DC/UCF system is known to other nodes in the DC/UCF communications network.

### **OPERATING SYSTEM**

Identifies the name of the host operating system under which the DC/UCF system runs.

### **DESCRIPTION**

Identifies the system description defined to the dictionary.

### **DATE CREATED**

Identifies the date the system occurrence was added to the dictionary.

### **PREPARED BY**

Identifies the user who added the system occurrence to the dictionary.

### **DATE LAST USED**

Identifies the date the system occurrence was last accessed.

### **REVISED BY**

Identifies the user who last modified the system occurrence.

### **STORAGE ALLOCATION**

Describes the storage allocation parameters:

#### **CWA SIZE**

Identifies size, in kilobytes, of the Common Work Area (CWA).

#### **PROGRAM POOL(K)**

Identifies the amount of storage, in kilobytes, that is available for loading nonresident programs, subschemas, maps, database procedures.

#### **STORAGE CUSHION(K)**

Identifies the amount of storage, in kilobytes, that is available in the storage pool for currently executing tasks.

**STORAGE POOL(K)**

Identifies the amount of storage, in kilobytes, that is provided for subschema work areas, COBOL working storage sections, user variable storage, packet-data-movement buffers, and SPF work areas (SPF users only).

**REENTRANT POOL(K)**

Indicates the size, in kilobytes, of an optional secondary program pool reserved for reentrant programs and tables (for example, subschemas, database procedures, and IDMS-DC maps).

**XA PROGRAM POOL**

For systems supporting 31-bit addressing, identifies the size, in kilobytes, of the 31-bit program pool.

**XA STORAGE POOL**

For systems supporting 31-bit addressing, identifies the size, in kilobytes, of storage pool number 255 (the 31-bit storage pool).

**XA REENTRANT POOL**

For systems supporting 31-bit addressing, indicates the size, in kilobytes, of the 31-bit reentrant pool.

**RELOCATABLE THRESHOLD (%)**

Indicates that the system should write relocatable storage to the scratch area across a pseudo-converse when the amount of space used in the storage pool reaches the indicated percentage.

**RUNTIME ENVIRONMENT**

Describes the run-time parameters:

**MAXIMUM TASKS**

Indicates the maximum number of user tasks that can be active concurrently. This number does not include external request units or the IDMS-DC system tasks. For users executing under the central version, the value includes tasks invoked by online IDD, the online subschema compiler, and CA-OLQ.

**SYSTEM RUN UNITS**

Indicates the number of system run units initiated at startup to service:

Message dictionary requests (MSGDICT)

Queue requests (SCRATCH/QUEUE)

Signon requests (SIGNON/DEST)

Dictionary load requests (LOADER)

Security requests on system-level resources (SECURITY)

Destination requests (SYSTEM/DEST)

**STORAGE PROTECTION**

(IDMS-DC only) Indicates whether the storage protection feature is enabled for the system.

**ALTERNATE PROT KEY**

(IDMS-DC only) Indicates the number of the alternate storage protect key used by the IDMS-DC system storage protection feature.

**TICKER INTERVAL**

Indicates how frequently, in wall-clock seconds, the system checks for the occurrence of timer-related events.

**AUTOMATIC NEW COPY**

Indicates the action taken if, in loading a program from the load area, the system finds that the program has been deleted. YES indicates that the system attempts to load the program from the system load library. NO indicates that the system does not attempt to load the program from the system load library until a DCMT VARY PROGRAM NEW COPY or an IDMS-CV operator VARY SUBSCHEMA NEW COPY is issued.

**UNDEFINED LOADABLE**

Indicates whether SUBSCHEMAS, MAPS, TABLES, DIALOGS, and ACCESS MODULES can be automatically defined to the dictionary at run time.

**PRINTED REP RET**

(IDMS-DC only) Identifies the amount of time, in days, that the DC system retains a report in the scratch/queue area.

**PRINTER CHECKPOINT**

(IDMS-DC only) Identifies the page count (printer checkpoints) for all active reports. A value of 0 or OFF indicates that interrupted reports are reprinted from the beginning. Any other value indicates that printing is resumed at the last checkpoint.

**RESOURCE TIMEOUT INTERVAL**

(IDMS-DC only) Identifies the amount of time, in wall-clock seconds, that the IDMS-DC system permits a terminal to be inactive before freeing the terminal resources.

**RESOURCE TIMEOUT PROGRAM/VERSION**

(IDMS-DC only) Identifies the name and version number of the program invoked by IDMS-DC to handle the resources of an inactive terminal that has exceeded the timeout interval.

**OLQ**

Indicates whether the system includes CA-OLQ.

**ADSO**

Indicates whether the system includes CA-ADS.

**PF KEYS**

Indicates whether the system includes at least one keys table.

**OLM KEYS**

Indicates whether a key table is defined for OLM.

**PAGE REL**

Indicates whether the system invokes operating system services when one or more virtual pages are no longer required and the contents of those pages need not be saved.

**LOADLIST**

Identifies the default load list to be used by the system when searching for programs.

**XA SCRATCH**

Indicates whether or not the scratch area (DDLDCSCR) uses a 31-bit storage pool.

**INTERNAL LIMITS**

Indicates whether DC/UCF controls limits on all tasks defined to the system during system generation or at run time. ENABLED indicates that limits are enforced; DISABLED indicates that limits are not enforced except by means of a DCMT VARY LIMITS command. OFF indicates that limits are not enforced.

**STG LIMIT**

Indicates the storage limit, in kilobytes, that a task can hold at one time.

**LOCK LIMIT**

Indicates the limit of record locks that a task can set.

**CALL LIMIT**

Indicates the limit of system service calls (for example, #GETSTG, #LOAD) a task can issue.

**DBIO LIMIT**

Indicates the limit of database I/O operations (for example, reads and writes) that are performed for a task.

**EXTERNAL LIMITS**

Indicates whether DC/UCF controls limits on all tasks associated with external request units (that is, ERUS tasks). ENABLED indicates that limits are enforced; DISABLED indicates that limits are not enforced except by means of a DCMT VARY LIMITS command. OFF indicates that limits are not enforced.

**STG LIMIT**

Indicates the storage limit, in kilobytes, that a task can hold at one time.

**LOCK LIMIT**

Indicates the limit of record locks that a task can set.

**CALL LIMIT**

Indicates the limit of system service calls (for example, #GETSTG, #LOAD) a task can issue.

**DBIO LIMIT**

Indicates the limit of database I/O operations (for example, reads and writes) that are performed for a task.

**OPERATING SYSTEM INTERFACE**

Describes the parameters that define how the DC/UCF system and the operating system interact.

**LOG TO DATABASE**

Indicates whether log records are written to the DDLDCLOG area of the dictionary.

### **LOGGING INFORMATION**

Identifies the system log file and the maximum number of records that can be written to FILE 1, the primary file, and FILE 2, the alternate log file. If COUNT 1 is 0, a maximum number of records was not defined. If COUNT 2 is -1, an alternate log file was not defined. If COUNT 1 and COUNT 2 are both 0, log records are written to the DDLDCLOG area of the dictionary.

### **LOG DEVICE**

(VSE/ESA only) Identifies the device type for the log file.

### **SVC NUMBER**

Identifies the number of the SVC used for communication between IDMS-CV and external request units.

### **3270 PRINT KEY**

Identifies the PF key used to print screen contents.

### **WTO DESCRIPTION CODES**

(OS/390 only) Identifies the OS/390 operator message codes, which are used to identify the values supplied to the DESC parameter for write-to operator (WTO) macros issued by the system.

### **WTO ROUTE CODES**

(OS/390 only) Identifies the OS/390 operator message routing codes, which are used to identify values for the ROUTCDE parameter for write-to-operator (WTO) macros issued by the system.

### **ERROR HANDLING**

Describes error handling procedures.

### **INACTIVE INTERVAL**

(IDMS-DC only) Indicates the time, in wall-clock seconds, that the system allows an internal task to wait for a resource before abnormally terminating the task.

### **RUNAWAY INTERVAL**

Indicates the maximum time, in wall-clock seconds, that the system allows a task or transaction to execute between interval waits before abnormally terminating the task or transaction.

### **SYSTEM DUMP**

Indicates whether the DC/UCF system takes a memory dump for all system abend codes.

### **SYSTEM TRACE**

(IDMS-DC only) Indicates whether the IDMS-DC system trace facility is enabled to trace system events during program development and debugging.

### **TRACE ENTRIES**

(IDMS-DC only) Indicates the number of entries allocated to the trace table buffer.

### **USER TRACE**

(IDMS-DC only) Indicates whether the user trace facility is enabled to trace program requests for IDMS-DC system services.

**USERTRACE ENTRIES**

Indicates the number of entries allocated to the user trace buffer.

**ABRU SNAP**

Indicates whether a snap dump is written to the log when an external request unit terminates abnormally.

**DEADLOCK INTERVAL**

Indicates the amount of time, in wall-clock seconds, that elapses before the system searches for deadlocked tasks.

**SYSTEM INTERNALS**

Describe the internal characteristics of the system.

**ABEND STORAGE(WDS)**

(IDMS-DC only) Indicates the amount of storage, in fullwords, available to the IDMS-DC system for processing abends.

**DPE COUNT**

(IDMS-DC only) Indicates the number of deadlock prevention elements (DPEs) allocated to the IDMS-DC system at startup.

**ECB LIST(WDS)**

(IDMS-DC only) Indicates the size, in fullwords, of the storage allocated for the Event Control Block list.

**RCE COUNT**

(IDMS-DC only) Indicates the number of resource control elements (RCEs) allocated to the IDMS-DC system at startup.

**RLE COUNT**

Indicates the number of resource link elements (RLEs) allocated to the DC/UCF system at startup.

**STACKSIZE (WDS)**

(IDMS-DC only) Indicates the size, in fullwords, of the work storage stack within the task control element (TCE) of the IDMS-DC system.

**PRIMARY NULL PDES**

Indicates the number of null program definition elements (PDEs) allocated at system startup for the automatic definition of programs not defined in PROGRAM statements. The default 0 indicates that automatic definition is disallowed.

**SECONDARY NULL PDES**

Indicates the number of additional PDEs that can be allocated from the storage pool when the primary null PDEs have been used.

**STATISTICS OPTIONS**

Describe how system statistics are logged.

**TIME INTERVAL**

Indicates how frequently (in seconds) histograms and system statistics are collected.

**BY TASK**

Indicates whether the DC/UCF system collects CPU-time statistics for each task.

**BY TRAN**

Indicates whether the DC/UCF system collects statistics on a transaction-by-transaction basis for all tasks.

**BY LINE**

Specifies whether the DC/UCF system collects by-line histograms.

**BY USER**

Indicates whether the DC/UCF system collects separate by-task CPU time statistics for system-mode time and user-mode time.

**COLLECT/WRITE**

Indicates whether the system is collecting by-task histograms or task statistics. COLLECT (the default) indicates collection of by-task histograms; WRITE indicates collection of task statistics.

**DB PARAMETERS**

Describe database access, journaling, and locking parameters.

**JOURNAL RETRIEVAL**

Indicates whether the system is writing BGIN and ENDJ checkpoints to the journal file for retrieval transactions.

**MAXIMUM SYSLOCKS**

Indicates the maximum number of record that the system is to maintain for all run units at a given time.

**AREA THRESHOLD**

Indicates the point at which, during ready processing, the system will begin to accumulate area locks for a database transaction. OFF directs the system not to accumulate area locks until the system can acquire all areas needed by a database transaction.

**AREA RETRY**

Indicates the number of times the system will continue trying to gain access to all areas without accumulating area locks. FOREVER directs the system to keep trying until it acquires all areas or until operating system resource and time limits are exceeded.

**RETRIEVAL LOCKING**

Indicates whether the system is to maintain locks automatically for records in areas accessed in shared retrieval mode.

**UPDATE LOCKING**

Indicates whether the system is to maintain locks automatically for records in areas accessed in protected update mode.

**SCRATCH/QUEUE JOURNAL BEFORE ONLY**

Indicates whether the system performs partial journaling. NO indicates that full journaling is taking place (that is, both before and after images are being written to the journal file).



**JOURNAL FRAGMENT**

Indicates the maximum number of journal blocks to write to the journal file before the system writes a dummy segment (DSEG) record to the journal file. OFF indicates that the journal fragment interval is off.

**JOURNAL TRANSACTION**

Indicates the number of active transactions that must be running in a DC/UCF system to defer the writing of a journal block.

**CV PARAMETERS**

Describe abend handling and time parameters for the central version.

**CHKUSER TASKS**

(OS/390 only) Indicates the number of tasks started by the system at run time to detect abnormally terminated OS/390 batch transactions. The value also represents the maximum number of batch transactions that can access the database concurrently, overriding the MAXIMUM ERUS parameter.

**CVNUMBER**

Indicates the number of the DC/UCF system to the CA-IDMS SVC.

**EXTERNAL WAIT**

Indicates the time, in wall-clock seconds, that the system waits for an external request unit to issue a database request before abnormally terminating the transaction.

**INTERNAL WAIT**

Indicates the time, in wall-clock seconds, that the system permits an external request unit to wait for a database or system resource before abnormally terminating the transaction.

**MAXIMUM ERUS**

Indicates the maximum number of external request units (ERUs) that can be active concurrently.

**RUPRTY**

Indicates the default execution priority for all transactions.

**SYSCTL PARAMETERS**

Describes parameters for the system control file used by programs executing outside the system region/partition.

**DDNAME**

(OS/390 only) Identifies the ddname of the system control (SYSCTL) file used by the system.

**DBNAME**

Identifies the dictionary or database to which the system will route requests from programs using the SYSCTL file.

**ALWAYS/DEFAULT**

Indicates which database/dictionary is accessed by programs at run time. ALWAYS indicates that programs always use the database named in the SYSCTL file regardless of IDMSOPTI or program specifications. DEFAULT indicates that programs use the database/dictionary named in the SYSCTL file only if a

dictionary/database name is not specified in the IDMSOPTI module or in the program.

**NODE NAME**

Identifies the name of a DC/UCF system defined to DC/UCF communications network to which the system will route requests from programs using the SYSCTL file.

**ALWAYS/DEFAULT**

Indicates which node is accessed by programs at run time. ALWAYS indicates that programs use the node named in the SYSCTL file, regardless of the IDMSOPTI or program specifications. DEFAULT indicates that programs use the node named in the SYSCTL file only if a node is not specified in the IDMSOPTI module or in the program.

## 4.20 Task Description Reports (CREPORTs 005, 020, and 021)

### 4.20.1 Contents

Task description reports provide information about tasks that have been associated with a DC/UCF system. CREPORT 005 describes tasks associated with executable systems; these tasks are represented in the dictionary by the TASKLST-023 record. CREPORTs 020 and 021 describe tasks associated with systems that have been defined but not generated; these tasks are represented by the TASK-025 record.

The fields and format of the three task reports are almost identical. In CREPORTs 005 and 020, task/program relationships are listed in alphabetical order by task name. In CREPORT 020 task/program relationships are listed in alphabetical order by program name.

REPORT NO. 05 CREPORT 005					CA-IDMS/DC SYSTEM GENERATION REPORT SYSTEM NAME: DCSYSTEM SYSTEM VERSION: 99 TASK DESCRIPTION					REL 15.0		09/08/99 PAGE 1		
TASK NAME	TASK VERS	PROGRAM	PROG VERS	BUILDER	ACTION	TCF TASKCODE	TCF VERS	PRIORITY	INPUT	MAP	INTERNAL	DISABLED	SAVE SCREEN	STG LOC
ADAI	1	ADAPMAIN	1	R	U		1	100	YES	NO	NO	NO	YES	ANY
PROTOCOL		---RESOURCE TIMEOUT--- PROGRAM VERS INTERVAL			--AREA RETRY	ACQUIRE-- THRESHOLD	MAX TASKS	INACTIVE INTERVAL	- - - - - CALLS	- - - - - RESOURCE DBIO	LIMITS LOCKS	- - - - - STORAGE	PRODUCT CODE	PRINT KEY
DEFRESP		1 SYSTEM			SYSTEM	OFF	OFF	SYSTEM	SYSTEM	SYSTEM	SYSTEM	SYSTEM		SYS
TASK NAME	TASK VERS	PROGRAM	PROG VERS	BUILDER	ACTION	TCF TASKCODE	TCF VERS	PRIORITY	INPUT	MAP	INTERNAL	DISABLED	SAVE SCREEN	STG LOC
ADS	1	ADSORUN1	1	R	U		1	100	YES	NO	NO	NO	YES	ANY
PROTOCOL		---RESOURCE TIMEOUT--- PROGRAM VERS INTERVAL			--AREA RETRY	ACQUIRE-- THRESHOLD	MAX TASKS	INACTIVE INTERVAL	- - - - - CALLS	- - - - - RESOURCE DBIO	LIMITS LOCKS	- - - - - STORAGE	PRODUCT CODE	PRINT KEY
DEFRESP		1 SYSTEM			SYSTEM	SYSTEM	OFF	SYSTEM	SYSTEM	SYSTEM	SYSTEM	SYSTEM		SYS
TASK NAME	TASK VERS	PROGRAM	PROG VERS	BUILDER	ACTION	TCF TASKCODE	TCF VERS	PRIORITY	INPUT	MAP	INTERNAL	DISABLED	SAVE SCREEN	STG LOC

Figure 4-23. Sample CREPORT 005

### 4.20.2 Field descriptions

A description of the fields in the sample report follows:

#### SYSTEM NAME and VERSION

Identify the name and version of the system associated with the tasks being described.

**TASK and TASK VERS**

Identify the name and version number of the task being described.

**PROGRAM and PROG VERS**

Identify the name and version number of the initial program invoked by the system for the task being described.

**BUILDER**

Identifies the builder code for the record that defines the task occurrence. For a description of builder codes, see Table 4-1 on page 4-57 at the end of this chapter.

**ACTION**

Indicates whether the task occurrence has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate of the system.

**TCF TASKCODE**

Identifies the code that invokes the transfer control facility (TCF) control program (RHDCUMBR) under which this task will run.

**TCF VERS**

Identifies the version of the TCF task code.

**PRIORITY**

Identifies the dispatching priority for the task.

**INPUT**

Indicates whether the terminal input buffer associated with the task contains data in addition to the task code at run time.

**MAP**

Indicates whether a mapout operation is performed automatically when the task is invoked.

**INTERNAL**

Indicates whether the task can be invoked internally and/or externally. YES indicates that the task can only be invoked internally; NO indicates that the task can be invoked internally or externally.

**DISABLED**

Indicates whether the task is disabled when the system is started.

**SAVE SCREEN**

Indicates whether screen contents associated with the task are saved before an immediate-write data stream is written to the terminal.

**STG LOC**

Indicates whether programs that run under the task can reside anywhere in the DC/UCF region (ANY) or whether programs must reside below 16 megabytes (BELOW).

**PROTOCOL**

Indicates the response protocol to be used by the task when communicating with terminals associated with a VTAMLIN type line. Possible values are DEFRESP and EXPRESP.

**RESOURCE TIMEOUT PROGRAM/ VERS**

Identifies the name and version of the resource timeout program, which the system invokes to handle the resources owned by an inactive terminal following the expiration of the resource timeout interval.

**RESOURCE TIMEOUT INTERVAL**

Specifies the amount of time the system is to permit a terminal to be inactive before invoking the terminal resource program. SYSTEM directs the system to use the INTERVAL IS value specified on the RESOURCE TIMEOUT parameter of the SYSTEM system generation statement.

**AREA ACQUIRE RETRY**

Identifies the limit on the number of times the system will continue trying to gain access to all areas without accumulating area locks. FOREVER directs the system to keep trying until it acquires all areas or until operating system resource and time limits are exceeded. SYSTEM directs the system to use the value specified on the AREA ACQUISITION THRESHOLD parameter of the SYSTEM system generation statement.

**AREA ACQUIRE THRESHOLD**

Indicates the number of times, during ready processing, that the system will wait on an area lock before it starts to accumulate area locks for a transaction. OFF directs the system not to accumulate area locks. SYSTEM directs the system to use the values specified in the AREA ACQUISITION THRESHOLD parameter of the SYSTEM system generation statement.

**MAX TASKS**

Indicates the limit of maximum concurrent threads for a task. OFF indicates the system does not limit the number of concurrent threads.

**INACTIVE INTERVAL**

Indicates the amount of time the system is to permit an internal task to wait for a resource before abnormally terminating the task. SYSTEM directs the system to use the value specified on the INACTIVE INTERVAL parameter of the SYSTEM system generation statement.

**RESOURCE LIMIT CALLS**

Indicates the limit of system service calls (for example, OBTAIN CALC), that can be issued by an online task. SYSTEM directs the system to use the limit specified in the CALL LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

**RESOURCE LIMIT DBIO**

Indicates the limit of database I/O operations (reads and writes) that can be issued by an online task. SYSTEM directs the system to use the limit specified in the DBIO LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

**RESOURCE LIMIT LOCKS**

Indicates the limit of record locks allocated to an online task during the life of the task. SYSTEM directs the system to use the limit specified in the LOCK LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

**RESOURCE LIMIT STORAGE**

Indicates the limit of storage, in kilobytes, that an online task can hold at one time. SYSTEM directs the system to use the limit specified in the STORAGE LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

**PRODUCT CODE**

Indicates a generic name for a product and related task codes.

**PRINT KEY**

Indicates the print-screen key assignment. SYS directs the system to use the print key assignment specified on the PRINT KEY parameter of the SYSTEM system generation statement.

## 4.21 Builder codes

Table 4-1 presents the builder codes that are referenced in the DREPORTs and CREPORTs.

Table 4-1. Dictionary Builder Codes. The builder code identifies the component that defined the entity occurrence to the dictionary or the component that last updated the entity occurrence.

Builder Code	Input Source
D	DDDL compiler
S	Schema compiler
C	DC/UCF mapping compilers
V	Subschema compiler
G	DC/UCF system generation compiler, before GENERATE command
R	DC/UCF system generation compiler, after GENERATE command
M	DML processors
A	ADS/ONLINE dialog generator
X	IDMSDIRL utility





## Chapter 5. CA-ADS Reports — AREPORTS

---

5.1 Overview	5-3
5.2 Summary table	5-4
5.3 Uses for CA-ADS reports	5-5
5.4 Producing CA-ADS reports	5-6
5.4.1 Syntax	5-6
5.4.2 Parameters	5-6
5.4.3 Examples	5-7
5.5 AREPORT 001 and 002 — ADS Dialogs and Their Components	5-8
5.5.1 Contents	5-8
5.5.2 Field descriptions	5-9
5.6 AREPORT 003 — ADS Dialogs by Process Key	5-12
5.6.1 Contents	5-12
5.6.2 Field descriptions	5-12
5.7 AREPORT 004 — ADS Dialogs by Record Key	5-14
5.7.1 Contents	5-14
5.7.2 Field descriptions	5-14
5.8 AREPORT 005 — ADS Dialogs by Subschema Key	5-16
5.8.1 Contents	5-16
5.8.2 Field descriptions	5-16
5.9 AREPORT 006 — ADS Dialogs by Map Key	5-17
5.9.1 Contents	5-17
5.9.2 Field descriptions	5-17



## 5.1 Overview

CA-ADS reports list detail information about online or batch dialogs and their components, which include subschemas, maps, processes, work records, map records, database records, and logical records. Information about the dialogs and their components is stored in the data dictionary.

There are six available CA-ADS reports: AREPORTs 001 through 006. AREPORT 001 reports on all the dialogs defined to the data dictionary and their associated components. AREPORTs 002 through 005 are key reports, which document each dialog associated with a particular component.

## 5.2 Summary table

The table below lists the CA-ADS reports in order by report module number:

Table 5-1. AREPORT KEY parameters

<b>AREPORT Module</b>	<b>Report Title</b>	<b>KEY Parameter</b>
001	ADS Dialogs and Their Components (Detail)	
002	ADS Dialogs and Their Components (Key)	KEY PROG-NAME-051 'dialog-name'
003	ADS Dialogs by Process Key	KEY MOD-NAME-067 'process-name'
004	ADS Dialogs by Record Key	KEY RSYN-NAME-079 'record-name'
005	ADS Dialogs by Subschema Key	KEY SS-NAM-026 'subschema-name'
006	ADS Dialogs by Map Key	KEY MAP-NAME-098 'map-name'

## 5.3 Uses for CA-ADS reports

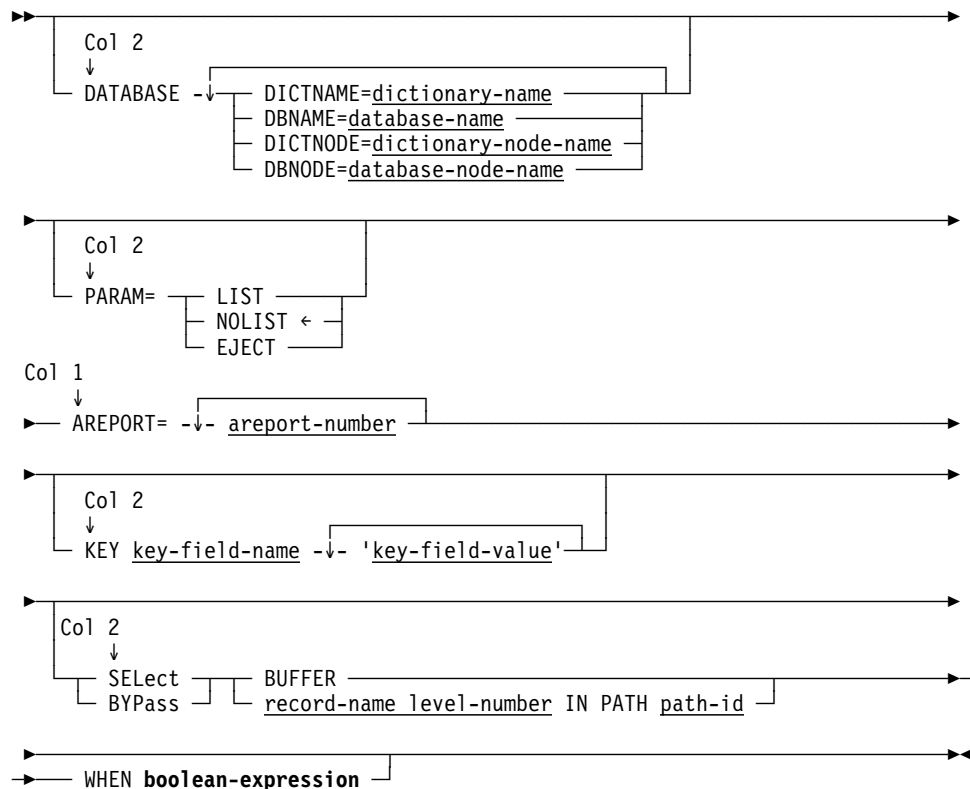
CA-ADS reports document dialogs defined to the data dictionary and identify associated components. CA-ADS reports can be used to:

- Assist in test to production migration (for example, AREPORT 002 identifies all the components associated with a particular dialog that must be migrated along with the dialog)
- Determine what dialogs have to be regenerated when a particular component changes (for example, AREPORT 005 lists all dialogs associated with a particular subschema)

## 5.4 Producing CA-ADS reports

CA-ADS reports are produced by submitting a job that includes the standard JCL shown in Appendices A through D and user-supplied control parameters. Syntax for the control parameters is shown below, followed by examples. Coding is freeform except for the starting column of each parameter.

### 5.4.1 Syntax



### 5.4.2 Parameters

Syntax rules appear in Chapter 1, "Introduction," except as described below:

#### **AREPORT=areport-number**

Identifies the report module name, where *areport-number* is a value in the range 001 through 006; leading zeros can be omitted. The parameter must be coded starting in column 1. Only one report module can be specified per run.

#### **KEY**

Specifies the parameter type, starting in column 2. A KEY parameter is required for AREPORTs 002 through 006. More than one KEY parameter can be specified per report.

**key-field-name**

Identifies the key field. Table 5-1 on page 5-4 lists possible values for each report.

**'key-field-value'**

Specifies a value or a list of values for the key field. Each value must be enclosed in single quotation marks. A list of items must be enclosed in parentheses and each item must be separated from another by a space or a comma.

### 5.4.3 Examples

**Example 1:** Data dictionary CULPDICT contains report module AREPORT 001, which is requested to report on dialogs defined to data dictionary DOCUDICT. The SELECT parameter selects only those dialogs that were created on September 8, 1999. By request, the input parameters will appear on the Sequential and Input Parameter Listings for the report.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
PARAM=LIST
AREPORT=001
SELECT BUFFER WHEN DATE-CREATED-051 EQ '09/08/99'
```

**Example 2:** AREPORT reports on all dialogs associated with the EMPLOYEE and DEPARTMENT records. The SELECT parameter selects only those dialogs that were created on September 8, 1999. By default, the input parameters will not appear on the report listings.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
AREPORT=004
KEY RSYN-NAME-079 ('EMPLOYEE' 'DEPARTMENT')
SELECT PROG-051 WHEN DATE-CREATED-051 EQ '09/08/99'
```

## 5.5 AREPORT 001 and 002 — ADS Dialogs and Their Components

### 5.5.1 Contents

The ADS Dialogs and Their Components reports list information about dialogs that have been generated within the DC/UCF system. AREPORT 001 is a detail report that lists every dialog; AREPORT 002 is a key report that lists selected dialogs. For each dialog, the reports provide information about the following components, if applicable:

- Subschema
- Map
- Processes
- Database records
- Logical records

The fields and format of the two reports are the same.

REPORT NO. 02	CA-ADS DIALOG REPORTER	R15.0	09/08/99 PAGE 1
CA-ADS DIALOGS AND THEIR COMPONENTS			
DIALOG...: LRTD01D	VERS.: 1	DATE CREATED: 03/29/99	DATE LAST UPDATED: 03/29/99 MAINLINE DIALOG
SUBSCHEMA NAME: EMPSSLR	SCHEMA NAME: EMPSCHM	SCHEMA VERSION: 1	
MAP: LRTD01M V 1	TYPE: ONL	DATE: 03/11/99 TIME: 153301 DDNAME-IN:	OUT: SUSP:
PROCESS NAME: LRTD01P	VERSION: 1	PROCESS TYPE: PREMAP	
DATE CREATED: 03/11/99	BY: MAD	DATE LAST UPDATED:	BY:
PROCESS NAME: LRTD01R-PF1	VERSION: 1	PROCESS TYPE: RESPONSE	
DATE CREATED: 03/11/99	BY: MAD	DATE LAST UPDATED:	BY:
CONTROL KEY/EVENT: PF1		RESPONSE FIELD VALUE:	
PROCESS NAME: LRTD01R-PF2	VERSION: 1	PROCESS TYPE: RESPONSE	
DATE CREATED: 03/11/99	BY: MAD	DATE LAST UPDATED:	BY:
CONTROL KEY/EVENT: PF2		RESPONSE FIELD VALUE:	
RECORD NAME: EMPLOYEE	VERSION: 1	*SUBSCHEMA*	*IN LR* *ONLMAP
RECORD NAME: ADSO-STAT-DEF-REC	VERSION: 1		*STATUS*
RECORD NAME: DEPARTMENT	VERSION: 1	*SUBSCHEMA*	*IN LR*
RECORD NAME: OFFICE	VERSION: 1	*SUBSCHEMA*	*IN LR*
RECORD NAME: EMPLOYEE-TABLE	SUBSCHEMA NAME: EMPSSLR	* LOGICAL RECORD *	

Figure 5-1. Sample AREPORT 002



## 5.5.2 Field descriptions

A description of the fields in AREPORTs 001 and 002 follows:

**DIALOG, VERS**

Name and version number of the dialog.

**DATE CREATED**

Date the dialog was added to the data dictionary.

**DATE LAST UPDATED**

Date the dialog was last modified.

**MAINLINE DIALOG**

A literal that appears if the dialog is a mainline dialog.

**SUBSCHEMA NAME**

Name of the subschema associated with the dialog. If no subschema is associated with the dialog, the following message appears:

\* \* NO SUBSCHEMA FOR THIS WK-DIALOG \* \*

**SCHEMA NAME, SCHEMA VERSION**

Name and version number of the schema associated with the subschema.

**MAP**

Name and version number of the map associated with the dialog. If no map is associated with the dialog, the following message appears:

\* \* MAPLESS DIALOG \* \*

**TYPE**

Indicates the type of map associated with the dialog:

- ONL is the dialog's online map.
- IN is the dialog's input file map.
- OUT is the dialog's output file map.

**DATE**

Date stamp for the map.

**TIME**

Time stamp for the map.

**DDNAME-IN**

(CA-ADS/Batch only) Indicates the ddname of an input file map.

**OUT**

(CA-ADS/Batch only) Indicates the ddname of an output file map.

**SUSP**

(CA-ADS/Batch only) Indicates the ddname of the suspense file associated with a dialog.

**MAP ENTRY PT**

Indicates that the map is the entry point for the dialog. This is true even if the dialog has a premap process.

**PROCESS NAME, VERSION**

Name and version number of a process associated with the dialog. If no processes are associated with the dialog, the following message appears:

\* \* NO PROCESSES FOR THIS WK-DIALOG \* \*

**PROCESS TYPE**

Indicates whether the process is used as a premap or response process for the dialog.

**EXECUTE ON EDIT ERRORS**

A literal that appears beside the response processes designated to execute even when there are map input errors.

**DATE CREATED, BY**

Date the process was added to the data dictionary and the ID of the user who created it.

**DATE LAST UPDATED, BY**

Date the process was last modified and the ID of the user who modified it.

**CONTROL KEY/EVENT**

Either of the following:

- (CA-ADS) The control key, if any, that causes the process to be executed if the process is a response process for the dialog
- (CA-ADS/Batch) A batch control event (EOF or IOERR), associated with an ADS/Batch dialog, that causes a process to be executed when its associated condition (end-of-file, I/O error) is met

**RESPONSE FIELD VALUE**

The response field value, if any, that causes the process to be executed if the process is a response process for the dialog.

**RECORD NAME, VERSION**

Name and version number of each record associated with the dialog. The following descriptors appear next to the records, as applicable:

- **\*SUBSCHEMA\*** indicates that the record is associated with the dialog as a subschema record.
- **\*WORK\*** indicates that the record is associated with the dialog as a work record.
- **\*N/C\*** indicates that the record is associated with the dialog as a new copy record.
- **\*STATUS\*** indicates that the record is associated with the dialog as a status definition record.
- **\*IN LR\*** indicates that the record is included in a logical record associated with the dialog.

- **\*INMAP\*, \*I/OMAP, \*ONLMAP, \*OUTMAP** indicate a record associated with a dialog's:

- Input map file
- Input and output map file
- Online map
- Output map file

If no records are associated with the dialog, the following message appears:

\* \* NO DATABASE RECORDS FOR THIS WK-DIALOG \* \*

**RECORD NAME, SUBSCHEMA NAME**

Record name and subschema name for each logical record associated with the dialog, along with a \* LOGICAL RECORD \* flag. If no logical records are associated with the dialog, the following message appears:

\* \* NO LOGICAL RECORDS FOR THIS WK-DIALOG \* \*

## 5.6 AREPORT 003 — ADS Dialogs by Process Key

### 5.6.1 Contents

AREPORT 003, the ADS Dialogs by Process Key report, lists all dialogs associated with one or more specified processes.

REPORT NO. 03			CA-ADS DIALOG REPORTER				R15.0		09/08/99 PAGE		1
CA-ADS DIALOGS BY PROCESS KEY											
PROCESS: LRTD01P			VERS.: 1		DATE CREATED: 03/11/99		BY: MAD		LAST UPDATED:		BY:
DIALOG NAME	DIALOG VERSION	DATE CREATED	DATE LAST UPDATED	MAINLINE DIALOG	PROCESS TYPE	CONTROL KEY/EVENT FOR PROCESS		RESPONSE FIELD VALUE FOR PROCESS		EXECUTE ON EDIT ERRORS	
LRTD01D	1	03/29/99	03/29/99	X	PREMAP						

Figure 5-2. Sample AREPORT 003

### 5.6.2 Field descriptions

A description of the fields in AREPORT 003 follows:

#### **PROCESS, VERS**

Name and version number of the process specified in the KEY parameter.

#### **DATE CREATED, BY**

Date the process was added to the data dictionary and the ID of the user who created it.

#### **LAST UPDATED, BY**

Date the process was last modified and the ID of the user who modified it.

#### **DIALOG NAME, DIALOG VERSION**

Name and version number of all dialogs associated with the process. If no dialogs are associated with the process specified in a KEY parameter, the following message appears:

\* \* NO DIALOGS USE THIS MODULE \* \*

#### **DATE CREATED**

Date the dialog was added to the data dictionary.

#### **DATE LAST UPDATED**

Date the dialog was last modified.

#### **MAINLINE DIALOG**

Indicates (with an X) whether the dialog is a mainline dialog.

#### **PROCESS TYPE**

Indicates whether the process is used as a premap or response process for the dialog.

**CONTROL KEY/EVENT FOR PROCESS**

Indicates either of the following:

- (CA-ADS) The control key, if any, that causes the process to be executed if the process is used as a response process for the dialog
- (CA-ADS/Batch) A batch control event (EOF or IOERR), associated with a CA-ADS/Batch dialog, that causes a process to be executed when its associated condition (end-of-file, I/O error) is met.

**RESPONSE FIELD VALUE FOR PROCESS**

The response field value, if any, that causes the process to be executed if the process is a response process for the dialog.

**EXECUTE ON EDIT ERRORS**

Indicates (with an X) whether a response process is designated to execute even when there are map input errors.

## 5.7 AREPORT 004 — ADS Dialogs by Record Key

### 5.7.1 Contents

AREPORT 004, the ADS Dialogs by Record Key report, lists all dialogs associated with one or more specified records.

REPORT NO. 04		CA-ADS DIALOG REPORTER					R15.0		09/08/99 PAGE 1		
CA-ADS DIALOGS BY RECORD KEY											
RECORD NAME...: EMPLOYEE							VERSION...: 1				
DIALOG NAME	DIALOG VERSION	DATE CREATED	DATE LAST UPDATED	MAINLINE DIALOG	MAP RECORD	SUBSCHEMA RECORD	WORK RECORD	NEW COPY RECORD	IN LOGICAL RECORD	STATUS RECORD	
ADMI01D	1	08/13/99	08/13/99	X	ONL	X					
DLRU01AD	1	04/16/99	04/16/99			X					
DLRU01D	1	07/16/99	07/16/99	X	ONL	X					
DRPP01D	1	07/29/99	07/29/99	X	ONL	X					
LDCT01D	1	08/16/99	08/16/99	X	ONL	X					
LDCT02D	1	08/16/99	08/16/99		ONL						
LDTD01AD	1	04/16/99	04/16/99			X					
LDTD01BD	1	04/16/99	04/16/99			X					
LDTD01D	1	07/23/99	07/23/99	X	ONL	X					
LNTN01D	1	05/07/99	05/07/99	X	ONL	X					
LNTN03D	1	05/07/99	05/07/99			X					
LRDA01D	1	03/23/99	03/23/99	X		X			X		
LRFC01D	1	03/30/99	03/30/99	X		X			X		
LRFC02D	1	03/30/99	03/30/99	X		X			X		
LRTD01D	1	03/29/99	03/29/99	X	ONL	X			X		

Figure 5-3. Sample AREPORT 004

### 5.7.2 Field descriptions

A description of the fields in AREPORT 004 follows:

#### RECORD NAME, VERSION

Name and version number of the subschema record, map record, or work record specified in the KEY parameter. If multiple versions of the record exist in the data dictionary, they are listed in ascending order.

#### DIALOG NAME, DIALOG VERSION

Names and version numbers of all dialogs associated with the record. If no dialogs are associated with a particular version of a record, the following message appears:

\* \* NO DIALOGS USE THIS RECORD \* \*

#### DATE CREATED

Date the dialog was added to the data dictionary.

#### DATE LAST UPDATED

Date the dialog was last modified.

#### MAINLINE DIALOG

Indicates (with an X) whether the dialog is a mainline dialog.

**MAP RECORD**

Indicates the type of map associated with the dialog:

- ONL is an online map.
- IN is an input map file.
- OUT is an output map file.
- I/O is an input and output map file.

**SUBSCHEMA RECORD**

Indicates (with an X) whether the record is associated with the dialog as a subschema record.

**WORK RECORD**

Indicates (with an X) whether the record is associated with the dialog as a work record.

**NEW COPY RECORD**

Indicates (with an X) whether the record is associated with the dialog as a new copy record.

**IN LOGICAL RECORD**

Indicates (with an X) whether the record is included in a logical record associated with the dialog.

**STATUS RECORD**

Indicates (with an X) whether the record is associated with the dialog as a status definition record.

## 5.8 AREPORT 005 — ADS Dialogs by Subschema Key

### 5.8.1 Contents

AREPORT 005, the ADS Dialogs by Subschema Key report, lists all dialogs associated with one or more specified subschemas.

REPORT NO. 05	CA-ADS DIALOG REPORTER			R15.0	09/08/99 PAGE 1
CA-ADS DIALOGS BY SUBSCHEMA KEY					
SUBSCHEMA NAME...: EMPSSLR	SCHEMA NAME...: EMPSCHM			SCHEMA VERSION...: 1	
DIALOG NAME	DIALOG VERSION	DATE CREATED	DATE LAST UPDATED	MAINLINE DIALOG	
OTFT03D	1	03/09/99	03/09/99		
REN001D	1	08/11/99	08/11/99	X	
REN002D	1	08/11/99	08/11/99	X	
REN003D	1	03/30/99	03/30/99	X	

Figure 5-4. Sample AREPORT 005

### 5.8.2 Field descriptions

A description of the fields in AREPORT 005 follows:

#### **SUBSCHEMA NAME**

Name of the subschema in the KEY parameter.

#### **SCHEMA NAME, SCHEMA VERSION**

Name and version number of the schema to which the subschema belongs.

#### **DIALOG NAME, DIALOG VERSION**

Names and version numbers of all dialogs associated with the subschema. If no dialogs are associated with the subschema named in the KEY parameter, the following message appears:

\* \* NO DIALOGS USE THIS SUBSCHEMA \* \*

#### **DATE CREATED**

Date the dialog was added to the data dictionary.

#### **DATE LAST UPDATED**

Date the dialog was last modified.

#### **MAINLINE DIALOG**

Indicates (with an X) whether the dialog is a mainline dialog.



## 5.9 AREPORT 006 — ADS Dialogs by Map Key

### 5.9.1 Contents

AREPORT 006, the ADS Dialogs by Map Key report, lists all dialogs associated with one or more specified maps.

REPORT NO. 06	CA-ADS DIALOG REPORTER	R15.0	09/08/99 PAGE 1
CA-ADS DIALOGS BY MAP KEY			
***** MAP NAME,,, LRTD01M	VERS: 1	MAP-DATE,,,,: 03/11/99	MAP-TIME,,,,: 153301
DIALOG...: LRTD01D	VERS: 1	MAP USE: ONL	DATE CREATED: 03/29/99
		LAST UPDATED: 03/29/99	***** MAINLINE DIALOG

Figure 5-5. Sample AREPORT 006

### 5.9.2 Field descriptions

A description of the fields in AREPORT 006 follows:

**MAP NAME, VERS**

Name and version number of the map specified in the KEY parameter.

**DIALOG, VERS**

Name and version number of a dialog associated with the map. If no dialogs are associated with the map named in the KEY parameter, the following message appears:

\* \* NO DIALOGS USE THIS MAP \* \*

**MAP-DATE**

Date stamp for the map.

**MAP-TIME**

Time stamp for the map.

**MAP USE**

Indicates the type of map associated with the dialog:

- ONL is an online map.
- IN is an input map file.
- OUT is an output map file.

**DATE CREATED**

Date the dialog was added to the data dictionary.

**LAST UPDATED**

Date the dialog was last modified.

**MAINLINE DIALOG**

A literal that appears if the dialog is a mainline dialog.



## Chapter 6. CA-IDMS/DB SQL Dictionary Reports — QREPORTS

---

6.1 Overview	6-3
6.2 Summary table	6-4
6.3 QREPORT 001 —SQL Column Name Report	6-5
6.3.1 Contents	6-5
6.3.2 Symbolic Parameter Overrides	6-5
6.3.3 Job Submission	6-5
6.3.4 Field descriptions	6-6
6.4 QREPORT 002 —Table and Column Report	6-8
6.4.1 Contents	6-8
6.4.2 Symbolic Parameter Overrides	6-8
6.4.3 Job Submission	6-8
6.4.4 Field descriptions	6-9
6.5 QREPORT 003 —Schema and Table Report	6-13
6.5.1 Contents	6-13
6.5.2 Symbolic Parameter Overrides	6-13
6.5.3 Job Submission	6-13
6.5.4 Field descriptions	6-14
6.6 QREPORT 004 —SQL Access Module Information	6-17
6.6.1 Contents	6-17
6.6.2 Symbolic Parameter Overrides	6-17
6.6.3 Job Submission	6-17
6.6.4 Field descriptions	6-18
6.7 QREPORT 005 —SQL Table Access Report	6-20
6.7.1 Contents	6-20
6.7.2 Symbolic Parameter Overrides	6-20
6.7.3 Job Submission	6-20
6.7.4 Field descriptions	6-21
6.8 QREPORT 006 —SQL Table Syntax Report	6-22
6.8.1 Contents	6-22
6.8.2 Symbolic Parameter Overrides	6-22
6.8.3 Job Submission	6-22
6.8.4 Field descriptions	6-24
6.9 QREPORT 007 —SQL Table Index Report	6-25
6.9.1 Contents	6-25
6.9.2 Symbolic Parameter Overrides	6-25
6.9.3 Job Submission	6-25
6.9.4 Field descriptions	6-26
6.10 QREPORT 008 —SQL Table Constraint Report	6-28
6.10.1 Contents	6-28
6.10.2 Symbolic Parameter Overrides	6-28
6.10.3 Job Submission	6-28
6.10.4 Field descriptions	6-29



## 6.1 Overview

CA-IDMS/DB SQL Dictionary Reports present information about entities defined to CA-IDMS/DB using SQL commands. These reports provide Database Administrators with detailed information about the contents of the SQL Dictionary, and allow application programmers to obtain selected information from the SQL Dictionary.

QREPORTS consist of CA-CULPRIT syntax that is stored in either a partitioned data set or as modules of language CULPRIT in the CA-IDMS/DB Integrated Data Dictionary. You can execute them, one at a time, by submitting a CA-CULPRIT job in batch. Because each QREPORT has its own unique SQL Select clause for database retrieval, each report must be run separately.

You submit QREPORTS using the USE parameter. Enclosing the report name in double quotes on a CA-CULPRIT USE statement has a very explicit meaning. The syntax **USE "QRPT001"** tells CA-CULPRIT to read the highest version of module QRPT001 with language CULPRIT from the CA-IDMS/DB IDD.

The syntax **USE QRPT001** (without double quotes) tells CA-CULPRIT to read the QRPT001 member from the partitioned dataset named by the CULLIB DD statement in the JCL.

The optional WITH VALUES clause of the USE syntax allows you to override the default values of symbolic parameters embedded in the QREPORT syntax. Since the reports are coded with the LIKE predicate, wildcards may be used when specifying these symbolic parameters. If wildcards are not used, however, the LIKE behaves as the equivalent of a comparison using an equal sign, with the restriction that the lengths of the two values being compared must be identical. Each QREPORT is documented with the symbolic parameters available and their lengths. For more information on the LIKE predicate, see the *CA-IDMS/DB SQL Reference*.

There are eight available CA-IDMS/DB Dictionary Reports: QREPORTs 001 through 008.

## 6.2 Summary table

The table below lists the QREPORTS in order by report module number:

---

Table 6-1. QREPORT summary table

---

<b>QREPORT Module</b>	<b>Report Title</b>
001	SQL Column Name Report
002	SQL Table Information Report
003	SQL Schema Information Report
004	SQL Access Module Information Report
005	SQL Table Access Report
006	SQL Table Syntax Report
007	SQL Table Index Report
008	SQL Table Constraint Report

---

## 6.3 QREPORT 001 —SQL Column Name Report

### 6.3.1 Contents

QREPORT 001, the **Column Name Report**, provides detailed information about each SQL column selected. The tabular report format lists the column names in alphabetical order.

QREPORT 001 is useful when you need to find out which tables contain a specific SQL column. It also lists the important attributes of each column.

### 6.3.2 Symbolic Parameter Overrides

**REQUESTED\_DICTIONARY (&&1)**

Specify the 1 to 8 character name of a specific dictionary.

**REQUESTED\_SCHEMA (&&2)**

Specify the 1 to 18 character name of a specific SQL schema.

**REQUESTED\_COLUMN (&&3)**

Specify the 1 to 32 character name of a specific column or use a *like-predicate pattern* to request more than one column.

### 6.3.3 Job Submission

This syntax will list all of the columns from the SYSTEM schema of the TSTDICT SQL dictionary.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT001" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='SYSTEM'
                           REQUESTED_COLUMN='%')
```

QREPORT 001	SQL COLUMN NAME REPORT						09/08/99	PAGE	3
COLUMN NAME	SCHEMA NAME / TABLE NAME	DATATYPE / NULL ATTR.	PREC.	SCALE	COLUMN / NULL OFFSET	INTERNAL LENGTH	DEFAULT VALUE	COL NUM IN TABLE	
NUMBER	SYSTEM RESOURCE	SMALLINT NOT NULL	15		130	2	NO	5	
NUMBLOCKS	SYSTEM JOURNAL	INTEGER NOT NULL	31		92	4	NO	9	
NUMBUFFERS	SYSTEM DMCL	SMALLINT NOT NULL	15		94	2	NO	9	
NUMCOLS	SYSTEM TABLE	SMALLINT NOT NULL	15		162	2	NO	26	
NUMCOLUMNS	SYSTEM CONSTRAINT	SMALLINT NOT NULL	15		116	2	NO	8	
NUMCOLUMNS	SYSTEM INDEX	SMALLINT NOT NULL	15		92	2	NO	8	
NUMDADS	SYSTEM SEGMENT	SMALLINT NOT NULL	15		96	2	NO	12	
NUMDBNAMES	SYSTEM DBTABLE	SMALLINT NOT NULL	15		62	2	NO	7	
NUMFILEMAPS	SYSTEM AREA	SMALLINT NULLS ALLOWED	15		96 95	2 1	NO	11	

Figure 6-1. Sample QREPORT 001

### 6.3.4 Field descriptions

A description of the fields in QREPORT 001 follows:

**COLUMN NAME**

The name of the column included in the schema table.

**SCHEMA NAME / TABLE NAME**

The schema and table name associated with the column.

**DATATYPE / NULL ATTR.**

The data type of the named column, and an indicator to specify whether the column can or cannot contain null values.

**PREC.**

The number of digits in a numeric value.

**SCALE**

The number of digits to the right of the decimal point in a numeric value.

**COLUMN / OFFSET**

The offset within a row to the data for the named column and the offset within a row to the null indicator value for the column. Offsets are relative to zero.

**NULL INTERNAL LENGTH**

The internal length of the data for the named column and the internal length of the null indicator value for the column.



**DEFAULT VALUE**

Indicates whether the WITH DEFAULT clause was specified for the column.

**COL NUM IN TABLE**

The relative position of the column in the table.

## 6.4 QREPORT 002 —Table and Column Report

### 6.4.1 Contents

QREPORT 002, the **Table and Column Report**, provides detailed information about each SQL Table selected, and also lists summary information about each column contained in the table.

This information is especially useful to application developers who need to know the names and attributes of each column in a table.

### 6.4.2 Symbolic Parameter Overrides

#### **REQUESTED\_DICTIONARY (&&1)**

Specify the 1 to 8 character name of a specific dictionary.

#### **REQUESTED\_SCHEMA (&&2)**

Specify the 1 to 18 character name of a specific SQL schema, or use a *like-predicate pattern* to request more than one schema.

#### **CREATED\_BY (&&3)**

Specify the 1 to 18 character name of a specific user or use a *like-predicate pattern* to request more than one user.

#### **REQUESTED\_TABLE (&&4)**

Specify the 1 to 18 character name of a specific table or use a *like-predicate pattern* to request more than one table.

### 6.4.3 Job Submission

The following syntax will produce an SQL Table Report for the SYSTEM.COLUMN table that resides in dictionary TSTDICT.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT002" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='SYSTEM'
                           CREATED_BY='%'
                           REQUESTED_TABLE='COLUMN')
```

QREPORT 002

SQL TABLE INFORMATION

09/08/99

PAGE 1

SCHEMA NAME: SYSTEM

TABLE NAME: COLUMN

TIMESTAMP: 1999-01-01-00.00

DATE CREATED: 1999-01-01-00.00

DATE LAST UPDATED: 1999-01-01-00.00

TABLE OWNER: MET

BY USER: MET

BY USER: MET

SEGMENT NAME: SYSCAT

AREA NAME: DDL CAT

TABLE ID: 1028

TABLE TYPE: TABLE

LOC. MODE: CALC

COMPRESS: NO

FORMAT: FIXED

UPDATABLE:

CHECKOPT:

TABLE LENGTH: 184

DATA LENGTH: 164

KEY LENGTH: 20

CONTROL LENGTH: 72

FIXED LENGTH: 164

SECTION LENGTH: 0

NUMBER OF SYNTAX: 0

DISPLACEMENT: 0

ESTIMATED ROWS: 0

NUMBER PAGES: 0

NUMBER ROWS: 0

ROWS PER PAGE: 0

NUMBER OF COLUMNS: 18

NUMBER OF INDEXES: 0

NUMBER OF TIMES REFERENCED: 0

REFERENCES TO OTHER TABLES: 2

COL  
NUM

COLUMN NAME

DATATYPE

PREC, SCALE

COLUMN  
OFFSET

COLUMN  
LENGTH

DEFAULT  
VALUE

NULL ATTR.

---

1 NAME

CHARACTER

0

32

NO

NOT NULL

2 NUMBER

SMALLINT

15

32

2

NO

NOT NULL

3 SCHEMA

CHARACTER

34

18

NO

NOT NULL

4 TABLE

CHARACTER

52

18

NO

NOT NULL

5 TYPE

CHARACTER

70

18

NO

NOT NULL

6 TYPECODE

SMALLINT

15

88

2

NO

NOT NULL

7 PRECISION

SMALLINT

15

90

2

NO

NOT NULL

8 SCALE

SMALLINT

15

92

2

NO

NOT NULL

9 NULLS

CHARACTER

94

1

NO

NOT NULL

10 DEFAULT

CHARACTER

95

1

NO

NOT NULL

11 VOFFSET

SMALLINT

15

96

2

NO

NOT NULL

12 VLENGTH

SMALLINT

15

98

2

NO

NOT NULL

13 NOFFSET

SMALLINT

15

100

2

NO

NOT NULL

14 NLENGTH

SMALLINT

15

102

2

NO

NOT NULL

15 NUMVALUES

INTEGER

31

104

4

NO

NOT NULL

16 SECLWVAL

BINARY

108

8

NO

NOT NULL

17 SECHIGHVAL

BINARY

116

8

NO

NOT NULL

18 FILLER

BINARY

124

40

NO

NOT NULL

Figure 6-2. Sample QREPORT 002

## 6.4.4 Field descriptions

A description of the fields in QREPORT 002 follows:

### SCHEMA NAME

The name of the schema.

### TABLE NAME

The name of the table.

### TIMESTAMP

Table timestamp, used for synchronization with access module definitions.

### TABLE OWNER

The owner of the schema in which this table resides.

### DATE CREATED, BY USER

The userid of the person who submitted the SQL CREATE TABLE DDL, and the date and time the table was created.

**DATE LAST UPDATED, BY USER**

The userid of the person who last altered the table definition, and the date and time it occurred.

**SEGMENT NAME**

Name of the segment associated with the area where table rows are stored.

**TABLE ID**

Internal table identifier which identifies the rows of the table within an area.

**TABLE LENGTH**

The total length of a row of tables.

**AREA NAME**

Name of area where table rows are stored.

**TABLE TYPE**

The type of table:

- 'N' - Record in a non-SQL defined schema
- 'T' - Base table
- 'V' - View

**DATA LENGTH**

Internal length of the data portion of a table row (including 4-byte RDW for a compressed table).

**LOC. MODE**

The possible loc. modes are:

- CALC
- DIRECT
- VIA

**KEY LENGTH**

Internal length of the non-data portion (the prefix length) of a table row.

**PUT ROUTINE**

CA-IDMS Presspack data characteristic table (DCT) name.

**COMPRESS**

Compression indicator:

- 'N' - Uncompressed
- 'P' - Compressed with CA-IDMS Presspack

**CONTROL LENGTH**

Internal length of the control portion (without the prefix) of a table row.

**FORMAT**

Format of the table row:

- 'F' - Fixed length
- 'V' - Variable length (compressed tables only)

**FIXED LENGTH**

Internal length of the fixed portion (without the prefix) of the table row.

**UPDATABLE**

When TYPE is 'V', updatable view indicator:

- 'Y' - Updatable
- 'N' - Not updatable

**SECTION LENGTH**

Length of the I-tree stored in the associated section table rows.

**CHECKOPT**

When TYPE is 'V', WITH CHECK OPTION indicator:

- 'Y' - View defined with WITH CHECK OPTION
- 'N' - View defined without WITH CHECK OPTION

**NUMBER OF SYNTAX**

Number of the "create view" or "create/alter table" DDL syntax records stored in the SYNTAX table. Create and alter table syntax is present only if it contains a CHECK clause.

**NUMBER OF COLUMNS**

Number of columns in the table or view.

**DISPLACEMENT**

Displacement, in pages, from clustering index or referenced row in a clustering constraint.

**NUMBER OF INDEXES**

Number of indexes on the table.

**ESTIMATED ROWS**

Estimated number of rows in the table.

**NUMBER OF TIMES REFERENCED**

Number of constraints in which the table is the referenced table.

**NUMBER PAGES**

Number of pages containing rows of the table when statistics were last updated.

**REFERENCES TO OTHER TABLES**

Number of constraints where this table is the referencing table.

**NUMBER ROWS**

Actual number of rows in the table when statistics were last updated.

**ROWS PER PAGE**

Number of table rows per page when statistics were last updated.

**COL NUM**

The column number in the table.

**COLUMN NAME**

The relative position of the column in the table.

**DATATYPE**

The data type for the named column.

**PREC.**

The number of digits in a numeric value.

**SCALE**

The number of digits to the right of the decimal point in a numeric value.

**COLUMN OFFSET**

The offset within a row into the data value of the column.

**COLUMN LENGTH**

The internal length of the column data value.

**DEFAULT VALUE**

Indicates whether or not the WITH DEFAULT clause was specified for this column.

**NULL ATTR.**

Indicates whether or not nulls are allowed.

## 6.5 QREPORT 003 —Schema and Table Report

### 6.5.1 Contents

QREPORT 003, the **Schema and Table Report**, provides a one page detailed listing of each SQL schema that in a dictionary. It also includes summary information about each table that belongs to the schema.

Not all schemas have associated tables. Therefore, QREPORT 003 uses an SQL "outer join" to provide information about all SQL schemas in the dictionary.

### 6.5.2 Symbolic Parameter Overrides

#### **REQUESTED\_DICTIONARY (&&1)**

Enter the 1 to 8 character name of the dictionary you wish to report on.

#### **REQUESTED\_SCHEMA (&&2)**

Specify a 1 to 18 character schema name, or use a *like-predicate pattern* to request multiple schemas.

### 6.5.3 Job Submission

The example below reports on all schemas residing in the TESTDICT dictionary whose schema name begins with DEMO.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT003" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='DEMO%')
```

QREPORT 003

SQL SCHEMA INFORMATION

09/08/99

PAGE 1

SCHEMA NAME: DEMO

DATE CREATED: 1999-02-12-11.17

DATE LAST UPDATED:

SCHEMA OWNER: MET

BY USER: MET

BY USER:

SEGMENT NAME: USERDB

AREA NAME: EMP\_AREA

SCHEMA TYPE: RELATIONAL

NON-SQL NODE:

NON-SQL DICTNAME:

NON-SQL SCHEMA:

SCHEMA VERSION:

TABLE NAME	TABLE ID	TABLE TYPE	CREATED BY	TOTAL LENGTH	NUMBER COLUMNS	NUMBER INDEXES	TIMES REFERENCED	REFERENCES OTHER TABLES
ARG2	1039	TABLE	BRASC02	8	2	1	0	0
ARG3	1040	TABLE	BRASC02	8	2	1	0	0
BUDGET	1025	TABLE	MET	28	5	1	0	1
DBCSTAB1	1093	TABLE	DDK	24	3	1	0	0
DBCSTAB2	1092	TABLE	DDK	16	2	1	0	0
DDKDBCS1	1102	TABLE	DDK	24	3	1	0	0
DDKTAB1	1090	TABLE	DDK	16	2	1	0	0
DEPT	1024	TABLE	MET	76	2	2	2	0
EMPL	1024	TABLE	MET	80	5	2	3	1
JOB	1026	TABLE	MET	40	2	2	1	0
MANAGERS	1026	TABLE	MET	32	3	1	0	2
NEWTABLE	1101	TABLE	GSR	24	2	1	0	0
POSITION	1025	TABLE	MET	32	3	1	0	2
SCALARS	1027	TABLE	BRASC02	200	19	1	0	0
TAB1	1036	TABLE	DLB	36	2	1	0	0
UWLABEL	1057	TABLE	ALP	208	7	1	0	0

Figure 6-3. Sample QREPORT 003 (Page 1)

QREPORT 003	SQL SCHEMA INFORMATION	09/08/99	PAGE 3
SCHEMA NAME: DEMOPROJ			
DATE CREATED:	1999-08-13-10.10	SCHEMA OWNER: ALP	
DATE LAST UPDATED:		BY USER: ALP	
		BY USER:	
SEGMENT NAME:		SCHEMA TYPE: RELATIONAL	
AREA NAME:			
NON-SQL NODE:		NON-SQL SCHEMA:	
NON-SQL DICTNAME:		SCHEMA VERSION:	
THIS SCHEMA HAS NO TABLES CURRENTLY DEFINED			

Figure 6-4. Sample QREPORT 003 (Page 3)

## 6.5.4 Field descriptions

A description of the fields in QREPORT 003 follows:

### SCHEMA NAME

Identifies the name of the schema.

### SCHEMA OWNER

Identifies the owner of the schema.



**DATE CREATED, BY USER**

Identifies the date and time the schema was created, and the user who created it.

**DATE LAST UPDATED, BY USER**

Identifies the date and time the schema was last updated, and the user who last updated it.

**SEGMENT NAME**

Name of the segment associated with the default area for relational schemas. For non-SQL schemas, it is the name of the database or segment containing the data that the non-SQL schema defines.

**AREA NAME**

Name of the default area for relational schemas.

**SCHEMA TYPE**

The type of schema:

- RELATIONAL
- NON SQL

**NON-SQL NODE**

Reserved for future use.

**NON-SQL SCHEMA**

Name of the non-SQL schema.

**NON-SQL DICTNAME**

Name of the dictionary in which the non-SQL schema is defined.

**SCHEMA VERSION**

Version number of the non-SQL schema.

**TABLE NAME**

Name of the table.

**TABLE ID**

Internal table identifier which identifies the rows of the table within an area.

**TABLE TYPE**

The type of table:

- NON SQL
- TABLE
- VIEW

**CREATED BY**

Identifies the user who created the table.

**TOTAL LENGTH**

The total length of a table row.

**NUMBER COLUMNS**

Number of columns in the table.

**NUMBER INDEXES**

Number of indexes on the table.

**TIMES REFERENCED**

Number of constraints in which the table is the referenced table.

**REFERENCES OTHER TABLES**

Number of constraints where this table is the referencing table.

## 6.6 QREPORT 004 —SQL Access Module Information

### 6.6.1 Contents

QREPORT 004, the **SQL Access Module Information** report, lists detail information about each access module defined in a dictionary, along with the names of all tables referenced by each access module. This report also uses an outer join to list access modules that do not reference any tables.

### 6.6.2 Symbolic Parameter Overrides

#### **REQUESTED\_DICTIONARY (&&1)**

Enter the 1 to 8 character name of the dictionary you wish to report on.

#### **REQUESTED\_SCHEMA (&&2)**

Enter the 1 to 18 character name of an SQL schema or use a *like-predicate pattern* to specify multiple schemas.

#### **REQUESTED\_AM (&&3)**

Enter the 1 to 18 character name of an access module, or use a *like-predicate pattern* to specify multiple access modules.

### 6.6.3 Job Submission

The example below reports on all Access Modules defined for the JMA schema in TSTDICT.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT004" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='JMA'
                           REQUESTED_AM='%')
```

QREPORT 004			SQL ACCESS MODULE INFORMATION		09/08/99	PAGE	1
AM NAME	VERSION	SCHEMA	TIME OF CREATION	LENGTH	TABLES ACCESSED		
					SCHEMA NAME	TABLE NAME	
DYNDIAL1	1	JMA	1999-06-18-17.03	0	DEMO	TAB1	
EXTCURS	1	JMA	1999-06-21-09.27	0	DEMO	TAB1	
EXTCURSA	1	JMA	1999-06-21-17.19	0	DEMO	TAB1	
JMAAUG01	1	JMA	1999-08-02-13.24	4,900	EMPDEMO	EMPLOYEE	
JMABAT02	1	JMA	1999-08-27-12.53	4,900	EMPDEMO	EMPLOYEE	
JMABAT03	1	JMA	1999-08-27-12.59	4,900	EMPDEMO	EMPLOYEE	
JMADYN1	1	JMA	1999-07-01-15.25	0	DEMO	TAB1	
JMANEST	1	JMA	1999-07-01-14.19	0	EMPDEMO	EMPLOYEE	
JMAPREP	1	JMA	1999-08-10-11.11	4,800	EMPDEMO	EMPLOYEE	
JMAPREP4	1	JMA	1999-04-23-11.15	0	EMPDEMO	EMPLOYEE	
JMASQL	1	JMA	1999-06-24-16.22	0	EMPDEMO	EMPLOYEE	
JMASQLB	1	JMA	1999-09-07-14.05	4,900	EMPDEMO	EMPLOYEE	
JMASQLD	1	JMA	1999-06-21-15.02	0	EMPDEMO	EMPLOYEE	
JMASQLIV	1	JMA	1999-06-28-09.48	0	INV	PART	
JMASQLI2	1	JMA	1999-06-28-09.27	0	INV	COMPONENT	
					INV	PART	
JMASQL2	1	JMA	1999-08-20-11.19	4,900	EMPDEMO	EMPLOYEE	
JMASQL2A	1	JMA	1999-09-03-09.53	4,900	EMPDEMO	EMPLOYEE	
JMASQL30	1	JMA	1999-07-30-10.38	4,900	EMPDEMO	EMPLOYEE	
PREPARE	1	JMA	1999-04-16-14.46	0			
TSTSQL	1	JMA	1999-07-19-17.52	0	EMPDEMO	EMPLOYEE	
TSTSQLAM	1	JMA	1999-07-19-17.54	0	EMPDEMO	EMPLOYEE	

Figure 6-5. Sample QREPORT 004

## 6.6.4 Field descriptions

A description of the fields in QREPORT 004 follows:

### AM NAME

Specifies the name of the access module.

### VERSION

Specifies the version number of the access module.

### SCHEMA

Specifies the schema associated with the access module.

### TIME OF CREATION

Specifies the time and date the access module was created.

**LENGTH**

Number of bytes in the access module.

**TABLES ACCESSED**

Identifies the tables accessed by this access module.

**SCHEMA NAME**

Specifies the name of the schema to which the table being accessed belongs.

**TABLE NAME**

Specifies the name of the table being accessed.

## 6.7 QREPORT 005 —SQL Table Access Report

### 6.7.1 Contents

QREPORT 005, the **SQL Table Access** report, lists each SQL table referenced by an access module, along with the names of all access modules that reference it. The report is sorted by schema name, table name, access module name, and access module version.

Tables with no referencing access modules are omitted from this report.

### 6.7.2 Symbolic Parameter Overrides

#### **REQUESTED\_DICTIONARY (&&1)**

Enter the 1 to 8 character name of the dictionary you wish to report on.

#### **REQUESTED\_SCHEMA (&&2)**

Enter the 1 to 18 character name of an SQL schema, or use a *like-predicate pattern* to request more than one schema.

#### **REQUESTED\_TABLE (&&3)**

Enter the 1 to 18 character name of an SQL table, or use a *like-predicate pattern* to request more than one table.

### 6.7.3 Job Submission

The example below lists all tables in all schemas of the TSTDICT dictionary that have referencing access modules.

The use of the DICTNAME clause on the DATABASE parameter allows CA-CULPRIT to retrieve QRPT005 syntax from the DIRLDICT dictionary.

```
DATABASE DICTNAME=dirldict
USE "QRPT005" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='% '
                           REQUESTED_TABLE='% ' )
```

QREPORT 005		SQL TABLE ACCESS REPORT		09/08/99	PAGE	1
SCHEMA.TABLE NAME		ACCESS MODULES THAT REFERENCE TABLE				
		AM NAME	VERSION	AM SCHEMA		
DEMO.TAB1		DYNDIAL1	1	JMA		
		EXTCURS	1	JMA		
		EXTCURSA	1	JMA		
		INSBDIAL	1	DEMO		
		JMADYN1	1	JMA		
DEMOEMPL.DBCSTAB2		DBCSDIA2	1	DEMOEMPL		
DEMOEMPL.DBCSTAB3		DBCSDIA5	1	DEMOEMPL		
DEMOEMPL.DIVISION		AMDCSQLT	1	DEMOEMPL		
EMPDEMO.EMPLOYEE		JMAAUG01	1	JMA		
		JMABAT02	1	JMA		
		JMABAT03	1	JMA		
		JMANEST	1	JMA		
		JMAPREP	1	JMA		
		JMAPREP4	1	JMA		
		JMASQL	1	JMA		
		JMASQLB	1	JMA		
		JMASQLD	1	JMA		
		JMASQL2	1	JMA		
		JMASQL2A	1	JMA		
		JMASQL30	1	JMA		
		SQLDEMOD	1			
		TSTSQL	1	JMA		
		TSTSQLAM	1	JMA		
INV.COMPONENT		JMASQLI2	1	JMA		
INV.PART		JMASQLIV	1	JMA		
		JMASQLI2	1	JMA		

Figure 6-6. Sample QREPORT 005

## 6.7.4 Field descriptions

A description of the fields in QREPORT 005 follows:

### SCHEMA.TABLE NAME

The name of the schema and the table.

### ACCESS MODULES THAT REFERENCE TABLE

Specifies the AM Name, Version, and AM Schema used to reference the named table.

### AM NAME

Specifies the name of the access module.

### VERSION

Specifies the version number of the access module.

### AM SCHEMA

Specifies the schema that owns the access module.

## 6.8 QREPORT 006 —SQL Table Syntax Report

### 6.8.1 Contents

QREPORT 006, the **SQL Table Syntax Report**, lists the syntax used to create or alter check constraints within a table. It also lists the SQL "Create View" syntax for every view requested. Base tables without check constraints do not appear on this report.

### 6.8.2 Symbolic Parameter Overrides

**REQUESTED\_DICTIONARY (&&1)**

Enter the 1 to 8 character name of the SQL dictionary you wish to report on.

**REQUESTED\_SCHEMA (&&2)**

Enter the 1 to 18 character name of an SQL schema, or use a *like-predicate pattern* to request more than one schema.

**REQUESTED\_TABLE (&&3)**

Enter the 1 to 18 character name of an SQL table, or use a *like-predicate pattern* to request more than one table.

### 6.8.3 Job Submission

The example below lists syntax for all qualifying tables and views in the TSTDICT dictionary.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT006" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='% '
                           REQUESTED_TABLE='% ')
```



QREPORT 006	SQL TABLE SYNTAX REPORT	09/08/99	PAGE 1
<u>SCHEMA NAME</u>	<u>TABLE NAME</u>	<u>SYNTAX</u>	
DEMOEMPL	EMPLOYEE	<pre> CREATE TABLE      EMPLOYEE (EMP_ID            UNSIGNED NUMERIC(4,0)      NOT NULL,  MANAGER_ID        UNSIGNED NUMERIC(4,0)      ,  EMP_FNAME          CHAR(20)                  NOT NULL,  EMP_LNAME          CHAR(20)                  NOT NULL,  DEPT_ID            UNSIGNED NUMERIC(4,0)      NOT NULL,  STREET             CHAR(40)                  NOT NULL,  CITY               CHAR(20)                  NOT NULL,  STATE              CHAR(02)                  NOT NULL,  ZIP_CODE           CHAR(09)                  NOT NULL,  PHONE              CHAR(10)                  ,  STATUS             CHAR                      NOT NULL,  SS_NUMBER          UNSIGNED NUMERIC(9,0)      NOT NULL,  START_DATE         DATE                      NOT NULL,  TERMINATION_DATE   DATE                      ,  BIRTH_DATE         DATE                      , CHECK ( ( EMP_ID &lt;= 8999 ) AND (STATUS IN ('A', 'S', 'L', 'T') ) ) IN USERDB.EMP_AREA </pre>	
DEMOEMPL	POSITION	<pre> CREATE TABLE      POSITION (EMP_ID            UNSIGNED NUMERIC(4,0)      NOT NULL,  JOB_ID             UNSIGNED NUMERIC(4,0)      NOT NULL,  START_DATE         DATE                      NOT NULL,  FINISH_DATE        DATE                      ,  HOURLY_RATE        UNSIGNED DECIMAL(7,2)      ,  SALARY_AMOUNT      UNSIGNED DECIMAL(10,2)     ,  BONUS_PERCENT      UNSIGNED DECIMAL(7,3)      ,  COMM_PERCENT       UNSIGNED DECIMAL(7,3)      ,  OVERTIME_RATE      UNSIGNED DECIMAL(5,2)      , CHECK ( (HOURLY_RATE IS NOT NULL AND SALARY_AMOUNT IS NULL) OR (HOURLY_RATE IS NULL AND SALARY_AMOUNT IS NOT NULL) ) ) IN USERDB.EMP_AREA </pre>	
SYSCA	ACCESSIBLE_TABLES	<pre> CREATE VIEW SYSCA.ACCESSIBLE_TABLES (SCHEMA, TABLE, TYPE) AS SELECT SCHEMA, NAME, TYPE FROM SYSTEM.TABLE WHERE TYPE &lt;&gt; 'N' AND ACCESS(SCHEMA, NAME) = 'Y' UNION SELECT NAME, SUBSTR(RSYN_NAME_079, 1, 18), 'N' FROM SYSTEM.SCHEMA, SYSDICT."S-010", SYSDICT."SRCD-113", SYSDICT."RCDSYN-079" WHERE DICTIONARY = CURRENT DATABASE AND NTWKSHEMA &lt;&gt; ' ' AND S_NAM_010 = NTWKSHEMA AND S_SER_010 = VERSION AND RSYN_NAME_079 NOT BETWEEN 'SR0' AND 'SR9' AND ACCESS(NAME, RSYN_NAME_079) = 'Y' AND "S-SRCD" AND "RCDSYN-SRCD" </pre>	

Figure 6-7. Sample QREPORT 006

## 6.8.4 Field descriptions

A description of the fields in QREPORT 006 follows:

**SCHEMA NAME**

The schema name associated with the table.

**TABLE NAME**

The name of the table.

**SYNTAX**

Specifies the syntax used to create each SQL table defined with a check constraint, and each SQL view.

## 6.9 QREPORT 007 —SQL Table Index Report

### 6.9.1 Contents

QREPORT 007, the **SQL Table Index** report, provides information about each index that is associated with the schema and table requested. For requests involving multiple schemas and tables, the report is sorted by schema name, table name, and index name.

Indexes with a name of **HASH** represent calc keys defined on a base table using the SQL CREATE CALC key syntax. On the final release of QRPT007, the word CALC will appear in column 9 instead of INDEX HASH.

The report layout is generated in the form of syntax which is similar to that used to create the index.

### 6.9.2 Symbolic Parameter Overrides

#### **REQUESTED\_DICTIONARY (&&1)**

Enter the 1 to 8 character name of the SQL dictionary you wish to report on.

#### **REQUESTED\_SCHEMA (&&2)**

Enter the 1 to 18 character name of an SQL schema, or use a *like-predicate pattern* to request more than one schema.

#### **REQUESTED\_TABLE (&&3)**

Enter the 1 to 18 character name of an SQL table, or use a *like-predicate pattern* to request more than one table.

### 6.9.3 Job Submission

The example below will retrieve QRPT007 syntax from the DIRLDICT dictionary of IDD and will provide an index report of all tables in the SYSTEM schema of the TSTDICT dictionary.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT007" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='SYSTEM'
                           REQUESTED_TABLE='%')
```

QREPORT 007		SQL TABLE INDEX REPORT		09/08/99	PAGE	1
		SCHEMA NAME: SYSTEM				
	INDEX NAME	SCHEMA.TABLE / INDEX COLUMNS	ORDER	OTHER SPECIFICATIONS		
UNIQUE INDEX	HASH	ON SYSTEM.AM (NAME VERSION	ASC, ASC)	UNCOMPRESSED CLUSTERED		
UNIQUE INDEX	HASH	ON SYSTEM.AREA (NAME SEGMENT	ASC, ASC)	UNCOMPRESSED CLUSTERED		
UNIQUE INDEX	IX-AREA	ON SYSTEM.AREA (NAME SEGMENT	ASC, ASC)	UNCOMPRESSED IN SYSCAT.DDL CATX INDEX BLOCK CONTAINS 10 KEYS DISPLACEMENT IS 3 PAGES		
UNIQUE INDEX	IX-BUFFER	ON SYSTEM.BUFFER (NAME DMCL	ASC, ASC)	UNCOMPRESSED IN SYSCAT.DDL CATX INDEX BLOCK CONTAINS 10 KEYS DISPLACEMENT IS 3 PAGES		
UNIQUE INDEX	IX-CONSTRAINT	ON SYSTEM.CONSTRAINT (SCHEMA NAME	ASC, ASC)	UNCOMPRESSED IN SYSCAT.DDL CATX INDEX BLOCK CONTAINS 10 KEYS DISPLACEMENT IS 3 PAGES		
UNIQUE INDEX	IX-DBNAME	ON SYSTEM.DBNAME (DBTABLE NAME	ASC, ASC)	UNCOMPRESSED IN SYSCAT.DDL CATX INDEX BLOCK CONTAINS 10 KEYS DISPLACEMENT IS 2 PAGES		
UNIQUE INDEX	HASH	ON SYSTEM.DBTABLE (NAME	ASC)			

Figure 6-8. Sample QREPORT 007

### 6.9.4 Field descriptions

A description of the fields in QREPORT 007 follows:

#### SCHEMA NAME

The schema which owns the indexes listed below.

#### INDEX NAME

The name of the index that was created.

#### SCHEMA.TABLE

The schema and table name for which this index is defined.

**INDEX COLUMNS**

Specifies the column(s) used to create the index key.

**ORDER**

Indicates the sort order of the values in the named column(s), ascending or descending.

**OTHER SPECIFICATIONS**

Lists other specifications used to create the index, such as compressed/uncompressed, clustered, displacement.

## 6.10 QREPORT 008 —SQL Table Constraint Report

### 6.10.1 Contents

QREPORT 008, the **SQL Table Constraint** report, lists all of the referential constraints in which each table participates. This report is sorted first by the schema and table names that were requested on the WITH VALUES clause, then the constraint relationship, and finally the constraint name. Constraints where the table being reported is the referencing table are listed first, followed by constraints where the table being reported is the referenced table.

Two tables are involved in each referential constraint. The dependent table is called the **referencing table**. Data values for certain columns of the referencing table are restricted to only those values that exist in corresponding columns of the **table being referenced**. These columns in the referencing table are often referred to as **foreign key columns**; they correspond to **referenced columns** in the referenced table.

Constraints where the table being reported is the referencing table are listed first in QREPORT 008. Finally, QREPORT 008 lists the constraints where the table being reported is the table being referenced.

### 6.10.2 Symbolic Parameter Overrides

#### REQUESTED\_DICTIONARY (&&1)

Enter the 1 to 8 character name of the SQL dictionary you wish to report on.

#### REQUESTED\_SCHEMA (&&2)

Enter the 1 to 18 character name of an SQL schema, or use a *like-predicate pattern* to request more than one schema.

#### REQUESTED\_TABLE (&&3)

Enter the 1 to 18 character name of an SQL table, or use a *like-predicate pattern* to request more than one table.

### 6.10.3 Job Submission

This example will retrieve QRPT008 syntax from the CULLIB copybook library, and produce a Table Constraint Report for those tables in SYSTEM schema of TSTDICT that begin with the letter **D**.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT008" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
                           REQUESTED_SCHEMA='SYSTEM'
                           REQUESTED_TABLE='D%')
```

QREPORT 008	SQL TABLE CONSTRAINT REPORT	09/08/99	PAGE 1
<u>TABLE NAME</u>	<u>CONSTRAINT NAME</u>	<u>TABLE REFERENCED</u>	<u>REFERENCING TABLE</u>
SYSTEM.DBNAME	DBTABLE-DBNAME DBNAME-DBSEGMENT DBNAME-DBSSC	SYSTEM.DBTABLE	SYSTEM.DBSEGMENT SYSTEM.DBSSC
SYSTEM.DBSEGMENT	DBNAME-DBSEGMENT	SYSTEM.DBNAME	
SYSTEM.DBSSC	DBNAME-DBSSC	SYSTEM.DBNAME	
SYSTEM.DBTABLE	DBTABLE-DBNAME		SYSTEM.DBNAME
SYSTEM.DMCL	DMCL-BUFFER DMCL-DMCLSEGMENT DMCL-JOURNAL		SYSTEM.BUFFER SYSTEM.DMCLSEGMENT SYSTEM.JOURNAL
SYSTEM.DMCLAREA	AREA-DMCLAREA DMCLSEG-DMCLAREA	SYSTEM.AREA SYSTEM.DMCLSEGMENT	
SYSTEM.DMCLFILE	DMCLSEG-DMCLFILE FILE-DMCLFILE	SYSTEM.DMCLSEGMENT SYSTEM.FILE	
SYSTEM.DMCLSEGMENT	DMCL-DMCLSEGMENT SEGMENT-DMCLSEG DMCLSEG-DMCLAREA DMCLSEG-DMCLFILE	SYSTEM.DMCL SYSTEM.SEGMENT	SYSTEM.DMCLAREA SYSTEM.DMCLFILE

Figure 6-9. Sample QREPORT 008

## 6.10.4 Field descriptions

A description of the fields in QREPORT 008 follows:

### TABLE NAME

The name of the table which participates in constraints.

### CONSTRAINT NAME

The name of the referential constraint.

### TABLE REFERENCED

Specifies the table being referenced.

### REFERENCING TABLE

Specifies the dependent table in the constraint.





# Chapter 7. ASF Row-Level Security Reports — IREPORTS

---

- 7.1 Overview . . . . . 7-3
- 7.2 Producing ASF row-level security reports . . . . . 7-4
  - 7.2.1 Steps . . . . . 7-4
  - 7.2.2 Syntax . . . . . 7-4
  - 7.2.3 Parameters . . . . . 7-4
  - 7.2.4 Summary table . . . . . 7-5
  - 7.2.5 Example . . . . . 7-5
- 7.3 ASF row-level security reports . . . . . 7-6



## 7.1 Overview

IREPORTs provide summary and detail information about ASF tables that enforce row-level security. For example, with these reports, you can determine the security name associated with the ASF table, the users that can access the table, and the criteria that determine which rows of the table a particular user is allowed to see.

►► For more information about row-level security, see *CA-ICMS System Administration* and the *CA-IDMS ASF User Guide*.

## 7.2 Producing ASF row-level security reports

### 7.2.1 Steps

To obtain summary and detail reports about tables that enforce row-level security, follow these steps:

1. By using ASF, obtain the required passkey and authority:
  - a. **Obtain a READ passkey** to the \$SRT-OST-CROSS-REFERENCE\$ table.
  - b. **Obtain authority** to access the \$SECURITY-RUNTIME-TABLE\$ table.

►► For information about assigning passkeys and authority, see the *CA-IDMS ASF User Guide*.
2. Run an IREPORT report, under the IDMS-DC/UCF central version, that contains the input parameters described below. The source code for the IREPORTs must be stored in the ASF dictionary as modules named IREPORT 001, IREPORT 002, and so on.

### 7.2.2 Syntax

Syntax and syntax rules for the CULPRIT parameters required to run row-level security reports appear below, followed by examples.

```
Col 2
↓
►►— DATABASE DICTNAME=asf-dictionary-name —————►
►— PROFILE USER=user-id — PW=password —————►
►— =COPY 'IREPORT 00n' —————►►
```

### 7.2.3 Parameters

**DICTNAME=asf-dictionary-name**

Specifies the name of the dictionary that contains the CULPRIT source modules and to which the \$SRT-OST-CROSS-REFERENCE\$ table is defined.

**PROFILE USER=user-id**

Specifies the ID of the user running the IREPORT. The user must have a READ passkey to the \$SRT-OST-CROSS-REFERENCE\$ table and must be authorized to access the \$SECURITY-RUNTIME-TABLE\$ table.

**PW=password**

Specifies the password of the user running the CULPRIT reports. The user must have a READ passkey to the \$SRT-OST-CROSS-REFERENCE\$ table and must be authorized to access the \$SECURITY-RUNTIME-TABLE\$ table.

**IREPORT 00n**

Specifies the report number as described in the summary table below.

## 7.2.4 Summary table

IREPORT	Title	Sorted by
IREPORT 001	Row Level Security Summary Report	Table name
IREPORT 002	Row Level Security Detail Report	Table name
IREPORT 003	Row Level Security Summary Report by User	User id
IREPORT 004	Row Level Security Detail Report by User	User id
IREPORT 005	Row Level Security Summary Report by Owner/Security Name	Security name within user id

## 7.2.5 Example

In this example, user BEAR has a READ passkey to the \$SRT-OST-CROSS-REFERENCE\$ table and authority to access the \$SECURITY-RUNTIME-TABLE\$ table. The user requests a report that lists detailed information about each table in ASFDICT that enforces row-level security.

```

DATABASE DICTNAME=ASFDICT
PROFILE USER=BEAR PW=CUB
=COPY 'IREPORT 002'

```

## 7.3 ASF row-level security reports

REPORT NO. 01 IREPORT 001	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY SUMMARY REPORT	09/08/99 PAGE 1
TABLE.: BUDGET2 OWNER.: QAL	TDN...: 247	SECURITY NAME: QALPASS
USER ----	WHERE CLAUSE -----	
QAD	JAN GT 10000	

Figure 7-1. Sample IREPORT 001

REPORT NO. 02 IREPORT 002	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY DETAIL REPORT	09/08/99 PAGE 1
TABLE.: BUDGET2 OWNER.: QAL	TDN...: 247	SECURITY NAME: QALPASS
USER.: QAD STATUS: (V) VALIDATED WHERE.: JAN GT 10000	GROUP.:	

Figure 7-2. Sample IREPORT 002

REPORT NO. 03 IREPORT 003	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY SUMMARY REPORT BY USER	09/08/99 PAGE 1
ROW LEVEL SECURITY ACCESS FOR USER.: QAC		SECURITY NAME: QALPASS
TABLE OWNER SECNAME TDN WHERE CLAUSE ----- DEMO JSS DEMO123 137 LNAME = 'ABC'		
REPORT NO. 03 IREPORT 003	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY SUMMARY REPORT BY USER	09/08/99 PAGE 2
ROW LEVEL SECURITY ACCESS FOR USER.: QUAD		SECURITY NAME: DEMO123
TABLE OWNER SECNAME TDN WHERE CLAUSE ----- BUDGET2 QAL QALPASS 247 JAN GT 10000 SMALLTAB QAL QALPASS2 260 ACCOUNT EQ '1'		
REPORT NO. 03 IREPORT 003	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY SUMMARY REPORT BY USER	09/08/99 PAGE 3
ROW LEVEL SECURITY ACCESS FOR USER.: QAG		SECURITY NAME: SECURROW
TABLE OWNER SECNAME TDN WHERE CLAUSE ----- ROW LEVEL QAP SECURROW 223 COL1 = 69		

Figure 7-3. Sample IREPORT 003

REPORT NO. 04 IREPORT 004	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY DETAIL REPORT BY USER	09/08/99	PAGE 2
ACCESS FOR USER.: QAD	GROUP:		
TABLE.: BUDGET2 OWNER.: QAL WHERE.: JAN GT 10000 STATUS: (V) VALIDATED	TDN: 247	SECNAME: QALPASS	
TABLE.: SMALLTAB OWNER.: QAL WHERE.: ACCOUNT EQ '1' STATUS: (V) VALIDATED	TDN: 260	SECNAME: QALPASS2	

Figure 7-4. Sample IREPORT 004

REPORT NO. 05 IREPORT 005	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY SUMMARY REPORT BY OWNER/SECURITY NAME	09/08/99	PAGE 1
OWNER.....: QAC SECURITY NAME.: DEM0123			
TABLE NAME	TDN	USER	WHERE CLAUSE
-----	----	---	-----
DEMO	137	QAC	LNAME = 'ABC'
REPORT NO. 05 IREPORT 005	INFORMATION CENTER MANAGEMENT SYSTEM ROW LEVEL SECURITY SUMMARY REPORT BY OWNER/SECURITY NAME	09/08/99	PAGE 2
OWNER.....: QAD SECURITY NAME.: QALPASS			
TABLE NAME	TDN	USER	WHERE CLAUSE
-----	----	---	-----
BUDGET2	247	QAD	JAN GT 10000

Figure 7-5. Sample IREPORT 005

**Field descriptions****TABLE**

Name of a table for which row-level security is established.

**OWNER**

Owner of the table.

**TDN**

Table definition number of the table.

**SECURITY NAME (SECNAME)**

Security name assigned to the table.

**USER**

Id of a user authorized to access certain rows of the table.

**WHERE**

Specifies criteria that define the data that the user is permitted to see. If a user has unlimited access to the table (that is, no WHERE clause exists), the following literal appears:

\*\*\* NO RESTRICTIONS \*\*\*

**STATUS**

Specifies whether the \$SECURITY-RUNTIME-TABLE\$ entry has been validated; that is, whether user, owner, security name, and WHERE criteria are entered correctly in the \$SECURITY-RUNTIME-TABLE\$ table. The report generates the following messages:

- (V) VALIDATED
- (I) REQUIRES VALIDATION
- (E) ERROR IN VALIDATION

**GROUP**

Name of the group that has authority to access the table.



## Chapter 8. CA-IDMS/DB Journal Reports — JREPORTS

---

8.1 Overview	8-3
8.2 Summary of journal reports	8-4
8.3 Uses for journal reports	8-5
8.4 Types of journal records	8-6
8.5 Summary of records required for journal reports	8-14
8.6 Producing journal reports	8-15
8.6.1 Uses archive journal file as input	8-15
8.6.2 Required CA-CULPRIT modules	8-15
8.6.3 Syntax	8-15
8.6.4 Parameters	8-16
8.6.5 Examples	8-17
8.6.6 Operating system considerations	8-18
8.7 JREPORT 001 — Transaction Summary	8-20
8.7.1 Purpose	8-20
8.7.2 Field descriptions	8-20
8.8 JREPORT 002 — Program Termination Statistics	8-22
8.8.1 Purpose	8-22
8.8.2 Field descriptions	8-23
8.9 JREPORT 003 — Program I/O Statistics	8-25
8.9.1 Purpose	8-25
8.9.2 Field descriptions	8-25
8.10 JREPORT 004 — Program Summary	8-27
8.10.1 Purpose	8-27
8.10.2 Field descriptions	8-27
8.11 JREPORT 005 — Detail Area/Transaction	8-30
8.11.1 Purpose	8-30
8.11.2 Field descriptions	8-30
8.12 JREPORT 006 — Detail Program/Area	8-32
8.12.1 Purpose	8-32
8.12.2 Field descriptions	8-32
8.13 JREPORT 007 — Area Summary	8-34
8.13.1 Purpose	8-34
8.13.2 Field descriptions	8-34
8.14 JREPORT 008 — Formatted Record Dump	8-36
8.14.1 Purpose	8-36
8.14.2 Field descriptions	8-37



## 8.1 Overview

When an application program accesses and updates a database, CA-IDMS/DB logs backup information to a journal file, which records program activity against the database. The journal file contains:

- A **time record** that describes when journal records are written
- **Checkpoints** that describe the status of transactions accessing the database
- **Journal record entries** that contain the before and after images of database records

The journal file also contains CKPT and USER records; however, these records are not used in journal reports.

## 8.2 Summary of journal reports

A journal report describes the contents of the journal file. The table below lists the journal reports in order by report module number.

<b>JREPORT Module</b>	<b>Report Title</b>
000	No listing (contains journal record field descriptions)
001	Transaction Summary
002	Program Termination Statistics
003	Program I/O Statistics
004	Program Summary
005	Detail Area/Transaction
006	Detail Program/Area
007	Area Summary
008	Formatted Record Dump

This chapter discusses types of journal records, report-specific input parameters required to process a journal report, and sample output.

►► For more information about journaling, see *CA-IDMS Database Administration*.

## 8.3 Uses for journal reports

Journal reports produce statistics that can be used to monitor CA-IDMS/DB database and DC/UCF system performance. As a DBA, journal statistics allow you to monitor the following performance features for each program that accesses the database:

- **Database page access statistics**, such as the number of pages written, read, and requested by each program and the number of records requested and made current of the transaction
- **CALC and VIA overflow statistics**, such as the number of CALC records stored on pages other than the target page
- **Variable-length record statistics**, such as the number of variable-length record fragments stored on pages other than the target page
- **Area usage statistics**, such as the name and usage modes of areas accessed by each application program

**Reports help you monitor performance:** By running journal reports frequently, a DBA can monitor trends in database and system performance.

**Other reporting facilities:** Other tools available to the DBA are the Database Analysis Utility (IDMSDBAN) and the BACKUP utility.

►► For more information about these utilities, see Chapter 11, "Other CA-IDMS Reporting Facilities."

## 8.4 Types of journal records

Journal reports use the following journal record types:

Record type	Description
TIME	<p>Time record</p> <p>TIME records the date and time the contents of the journal buffer are written to the journal file. A TIME record is created each time a journal buffer is initialized; however, the date and time fields contain binary zeros until the contents of the journal buffer are written to the journal file.</p>
BGIN	<p>Checkpoint</p> <p>BGIN checkpoints mark the beginning of a database transaction. They are written to the journal file when an application initiates a non-SQL database transaction if JOURNAL RETRIEVAL is specified on the SYSTEM system generation statement, or when the first update occurs for any transaction if NOJOURNAL RETRIEVAL is in effect.</p>
AREA	<p>Checkpoint</p> <p>AREA checkpoints record transaction access to a database area. One AREA checkpoint is written to the journal file for each area readied by an explicit DML READY statement or readied automatically by the DBMS.</p> <p>AREA checkpoints are written to the journal file as follows:</p> <ul style="list-style-type: none"><li>▪ <b>Under the central version</b>, AREA checkpoints are written to the journal file at the time of the first functional call issued by the application program.</li><li>▪ <b>In local mode</b>, AREA checkpoints for areas readied in update mode are written to the journal file as each READY statement is processed. AREA checkpoints for areas readied in retrieval mode are written to the journal file at the time of the first functional call.</li></ul>
COMT	<p>Checkpoint</p> <p>COMT checkpoints mark the end of one recovery unit and the beginning of another within a transaction. A COMT checkpoint is created when a CA-IDMS/DB application program issues a COMMIT or COMMIT WORK CONTINUE statement.</p>

Record type	Description
ENDJ	<p>Checkpoint</p> <p>ENDJ checkpoints mark the normal termination of a transaction. An ENDJ checkpoint is created when a CA-IDMS/DB application program issues a FINISH or COMMIT WORK statement.</p>
RTSV	<p>Checkpoint</p> <p>RTSV checkpoints mark an SQL statement whose updates must be rolled out. An RTSV checkpoint is written to the journal file whenever an error is encountered while processing an SQL statement and that statement has updated the database.</p>
ABRT	<p>Checkpoint</p> <p>ABRT checkpoints mark the abnormal termination of a transaction. An ABRT checkpoint is created when a CA-IDMS/DB application program issues a ROLLBACK or ROLLBACK WORK statement, or if running under the central version, when CA-IDMS/DB automatically recovers a failing transaction.</p> <p>When an application program issues a ROLLBACK CONTINUE statement, the ABRT checkpoint written to the journal file marks the termination of a recovery unit within a transaction. In this case, the ABRT checkpoint is followed immediately by a BGIN checkpoint for the same transaction. New AREA checkpoints are also written.</p>
BFOR	<p>Journal record entry</p> <p>BFOR entries record the before image of a database record. A BFOR entry is created when a CA-IDMS/DB application program issues a request to update information in the database. When a new record is stored, the BFOR entry contains a null before image, indicating the previous absence of the record in the database. When a BFOR entry contains a null before image, the record-image-length field in the entry is set to zero.</p>
AFTR	<p>Journal record entry</p> <p>AFTR entries record the after image of a database record. An AFTR entry is created when a CA-IDMS/DB application program issues a request to update information in the database. When an existing record is removed, the AFTR entry contains a null after image, indicating the deletion of the record from the database. When an AFTR entry contains a null after image, the record-image-length field in the entry is set to zero.</p>

### Special considerations for BFOR and AFTR entries

- BFOR and AFTR entries are also written when CA-IDMS/DB brings relocated records back to the home page or physically deletes logically deleted records. These actions can occur during *any* operation; therefore, BFOR and AFTR entries can appear for retrieval-only transactions that have readied the applicable area in update mode.
- Corresponding BFOR and AFTR entries are not necessarily contiguous in the journal file. For example, the BFOR record for one transaction may be immediately followed by an AFTR record for a different transaction. The BFOR and AFTR entries for a database record contain either the whole record or a portion of the record, as follows:
  - If the data portion of the record has been changed, the journal record entries contain the entire database record (that is, the dbkey, prefix, and data portions).
  - If only pointers in the record prefix have been changed, only the dbkey and prefix portions of the record are written to the journal file.
- BFOR and AFTR entries can also span journal blocks. A record that spans a journal block appears as two records with the second portion offset by the length of the first portion. For example, if 200 bytes remain in a journal block, a 500-byte BFOR record is journaled as one 200-byte and one 300-byte BFOR journal entry; the 300-byte entry has an offset of 200.

The figures below show the record layouts for the journal record types.

\*1=Retrieval 2=Update  
 \*\*1=Shared 2=Exclusive  
 3=Protected

#### TIME JOURNAL RECORD TYPE

	Field	Data Type	Length (in bytes)
0 1	Journal record length	binary	2 (Halfword)
2 3	Reserved (initialized to zeroes)	binary	2 (Halfword)
4 7	Journal record type (TIME)	alphanumeric	4
8 F	UTC date/time stamp	binary	8 (2 Fullwords)
10 17	Journal record sequence number	binary	8 (2 Fullwords)
18 1F	Journal record qualifier	binary	8 (2 Fullwords)
20	CV number (central version only)	binary	1
21	Journaling version	binary	1
22 23	Reserved	binary	2 (Halfword)
24 2B	date/time stamp	binary	8 (2 Fullwords)
2C 2D	Journal record length	binary	2 (Halfword)
2E 2F	Reserved (initialized to zeroes)	binary	2 (Halfword)



## BGIN JOURNAL RECORD TYPE

Field	Data Type	Length (in bytes)
0 1	Journal record length	binary 2 (Halfword)
2 3	Reserved (initialized to zeroes)	binary 2 (Halfword)
4 7	Journal record type (BGIN)	alphanumeric 4
8 F	UTC date/time stamp	binary 8 (2 Fullwords)
10 17	Journal record sequence number	binary 8 (2 Fullwords)
18 1F	Journal record qualifier	binary 8 (2 Fullwords)
20	CV number (central version only)	binary 1
21	Journaling version	binary 1
22 23	Reserved	binary 2 (Halfword)
24 27	Journal transaction ID	binary 4 (Fullword)
28 2B	Address of VIB	address 4
2C 33	Program name	alphanumeric 8
34 3B	date/time stamp	binary 8 (2 Fullwords)
3C 43	Reserved (initialized to zeroes)	binary 8 (2 Fullwords)
44 45	Transaction update quiesce level	binary 2 (Halfword)
46 47	Transaction quiesce level	binary 2 (Halfword)
48 4F	Local task ID for transaction or savepoint sequence number	binary 8 (2 Fullwords)
50 51	Journal record length	binary 2 (Halfword)
52 53	Reserved (initialized to zeroes)	binary 2 (Halfword)

## AREA JOURNAL RECORD TYPE

Field	Data Type	Length (in bytes)
0 1	Journal record length	binary 2 (Halfword)
2 3	Reserved (initialized to zeroes)	binary 2 (Halfword)
4 7	Journal record type (AREA)	alphanumeric 4
8 F	UTC date/time stamp	binary 8 (2 Fullwords)
10 17	Journal record sequence number	binary 8 (2 Fullwords)
18 1F	Journal record qualifier	binary 8 (2 Fullwords)
20	CV number (central version only)	binary 1
21	Journaling version	binary 1
22 23	Reserved	binary 2 (Halfword)
24 27	Journal transaction ID	binary 4 (Fullword)
28	Area type	binary 1
29	DBMS verb number	binary 1
2A 2B	Schema page group identifier	binary 2 (Halfword)
2C 3D	Area name	alphanumeric 18
3E	Area flag byte	binary 1
3F	Reserved	binary 1
40 43	Low page of area	binary 4 (Fullword)
44 47	High page of area	binary 4 (Fullword)
48 49	*OPEN mode	binary 2 (Halfword)
4A 4B	**OPEN access	binary 2 (Halfword)
4C 4D	Journal record length	binary 2 (Halfword)
4E 4F	Reserved (initialized to zeroes)	binary 2 (Halfword)

## 8.4 Types of journal records

### BFOR and AFTR JOURNAL RECORD TYPE

	Field	Data Type	Length (in bytes)
0 1	Journal record length	binary	2 (Halfword)
2 3	Reserved (initialized to zeroes)	binary	2 (Halfword)
4 7	Journal record type (BFOR or AFTR)	alphanumeric	4
8 F	UTC date/time stamp	binary	8 (2 fullwords)
10 17	Journal record sequence number	binary	8 (2 fullwords)
18 1F	Journal record qualifier	binary	8 (2 fullwords)
20	CV number (central version only)	binary	1
21	Journaling version	binary	1
22 23	Reserved	binary	2 (Halfword)
24 27	Journal transaction ID	binary	4 (Fullword)
28	Area type	binary	1
29	DBMS verb number	binary	1
2A 2B	Schema page group identifier	binary	2 (Halfword)
2C 2F	Schema DBKey format word	binary	4 (Fullword)
30 33	Database key	binary	4 (Fullword)
34 35	Record ID	binary	2 (Halfword)
36 37	Page displacement	binary	2 (Halfword)
38 39	Record image length	binary	2 (Halfword)
3A 3B	Prefix length	binary	2 (Halfword)
3C 3D	Offset into DBMS of call	binary	2 (Halfword)
3E 3F	Spanned record offset	binary	2 (Halfword)
40	User record (prefix and data)	variable	
n+1 n+2	Journal record length	binary	2 (Halfword)
n+3 n+4	Reserved (initialized to zeroes)	binary	2 (Halfword)

\*Central Version Only

## COMT, ENDJ, ABRT, RTSV, Journal Record Type

	Field	Data Type	Length (in bytes)
0 1	Journal record length	binary	2 (Halfword)
2 3	Reserved (initialized to zeroes)	binary	2 (Halfword)
4 7	Journal record type (BGIN)	alphanumeric	4
8 F	UTC date/time stamp	binary	8 (2 Fullwords)
10 17	Journal record sequence number	binary	8 (2 Fullwords)
18 1F	Journal record qualifier	binary	8 (2 Fullwords)
20	CV number (central version only)	binary	1
21	Journaling version	binary	1
22 23	Reserved	binary	2 (Halfword)
24 27	Journal transaction ID	binary	4 (Fullword)
28 2B	Address of VIB	address	4
2C 33	Program name	alphanumeric	8
34 3B	date/time stamp	binary	8 (2 Fullwords)
3C 43	Reserved (initialized to zeroes)	binary	8 (2 Fullwords)
44 45	Transaction update quiesce level	binary	2 (Halfword)
46 47	Transaction quiesce level	binary	2 (Halfword)
48 4F	Local task ID for transaction or savepoint sequence number	alphanumeric or binary	8
50 53	Number of user records updated	binary	4 (Fullword)
54 57	Number of pages read from database	binary	4 (Fullword)
58 5B	Number of pages written to database	binary	4 (Fullword)
5C 5F	Total pages requested by DBMS	binary	4 (Fullword)
60 63	Number of times calc record stored on target page	binary	4 (Fullword)
64 67	Number of times calc record not stored on target page	binary	4 (Fullword)
68 6B	Number of times via record stored on target page	binary	4 (Fullword)
6C 6F	Number of times via record not stored on target page	binary	4 (Fullword)
70 73	Number of records requested from DBMS	binary	4 (Fullword)
74 77	Number of current-of-transaction records	binary	4 (Fullword)
78 7B	Number of calls to DBMS	binary	4 (Fullword)
7C 7F	Number of record fragments stored	binary	4 (Fullword)
80 83	Number of record fragments returned to home page	binary	4 (Fullword)
84 87	*Total locks acquired	binary	4 (Fullword)
88 8B	*Number of shared DB-key locks held	binary	4 (Fullword)
8C 8F	*Number of non-shared DB-key locks held	binary	4 (Fullword)
90 93	Transaction ID	binary	4 (Fullword)
94 97	Task ID	binary	4 (Fullword)
98 9F	Local task ID for transaction	binary	8 (2 Fullwords)
A0 A3	Number of SR8 splits	binary	4 (Fullword)
A4 A7	Number of SR8 spawns	binary	4 (Fullword)
A8 AB	Number of SR8 records stored	binary	4 (Fullword)
AC AF	Number of SR8 records erased	binary	4 (Fullword)
B0 B3	Number of SR7 records stored	binary	4 (Fullword)
B4 B7	Number of SR7 records erased	binary	4 (Fullword)
B8 BB	Number of B-Tree searches	binary	4 (Fullword)
BC BF	Number of B-Tree levels searched	binary	4 (Fullword)
C0 C3	Number of orphans adopted	binary	2 (Halfword)
C4 C5	# index levels Searched - best case	binary	2 (Halfword)
C6 C7	# index levels searched - worst case	binary	2 (Halfword)
C8 C9	Journal record length	binary	2 (Halfword)
CA CB	Reserved (initialized to zeroes)	binary	2 (Halfword)

**Field definitions for JREPORT 000:** The journal record field definitions as they appear in the JREPORT 000 parameter module appear below. The parameters show the actual names (and synonyms) of the journal record fields.

REC\$0****	'J' REPORTS	IDMS	JOURNAL	FILE	CAGJF0	REC PARAMETERS
REC REC-LEN				1	2	1
REC TYPE				5	4	
REC GMT				9	8	
REC SEQ				17	8	1
REC QUAL				25	8	1
REC QUAL-CV				25	8	
REC QUAL-LOCAL				25	8	1
REC CVNO				33	1	

\$qualifier:  
\$CV: node  
\$Local: GMT

## 8.4 Types of journal records

---

```

REC VERS                      34  1
REC RSV1                      35  2      $Reserved
REC $ BGIN,COMT,ENDJ,ABRT,RTSV, Plus AREA,BFOR,AFTR RECORDS *****
REC TRANSACT-ID              37  4 1
REC TRANSACT-IDC             37  4
REC $ BGIN,COMT,ENDJ,ABRT,RTSV, Plus AREA RECORDS *****
REC VIB                      41  4 1
REC $ BGIN,COMT,ENDJ,ABRT,RTSV RECORDS*****
REC PROGRAM-NAME             45  8
REC DTESTAMP                 53  8
REC DTESTAMPX                53  8 1
REC OLDTIM                   61  8 2      $not used
REC UPDATE-QUIESE            69  2 1
REC QUIESCE                  71  2 1
REC SAVEPOINT-SEQ            73  8 1      $RTSV only.
REC TASK-ID                  73  8      $Local Task id.
REC LT-P1                    73  4
REC LT-P2                    77  4 1
REC $ COMT,ENDJ,ABRT RECORDS ONLY *****
REC UPDATED                  81  4 1
REC DB-STATS                 85 60
REC READ                     85  4 1
REC WRITTEN                   89  4 1
REC PAGE-REQUESTS            93  4 1
REC CALC-ON                   97  4 1
REC CALC-NOT-ON              101  4 1
REC VIA-ON                   105  4 1
REC VIA-NOT-ON               109  4 1
REC REC-REQUESTED            113  4 1
REC CURR-OF-TR               117  4 1
REC CALLS                    121  4 1
REC FGMT-STORED              125  4 1
REC FGMT-RETURNED            129  4 1
REC LOCKS-REQUESTED          133  4 1
REC SHARED-LOCKS-HELD        137  4 1
REC EXCLUSIVE-LOCKS-HELD     141  4 1
REC TR-ID-CV                 145  4 1      $TRANSACTION ID.
REC TASK-ID-CV               149  4 1      $TCE address.
REC LOCAL-TASK-ID-CV         153  8
REC LT-PART1                 153  4
REC LT-PART2                 157  4 1
REC IX-STATS                 161 40
REC IX-SR8-SPLITS            161  4 1
REC IX-SR8-SPAWNS            165  4 1
REC IX-SR8-STORED            169  4 1
REC IX-SR8-ERASED            173  4 1
REC IX-SR7-STORED            177  4 1
REC IX-SR7-ERASED            181  4 1
REC IX-BTREE-SEARCHES        185  4 1
REC IX-BTREE-LEVELS-SEARCHED 189  4 1
REC IX-ORPHANS-ADOPTED       193  4 1
REC IX-LEVELS-SEARCHED-BEST   197  2 1
REC IX-LEVELS-SEARCHED-WORST 199  2 1
REC $ TIME RECORD ONLY *****
REC T-DATESTMP              37  8 1
REC $ AREA RECORD ONLY *****
REC AREA                     45 18
REC RESERVED2                63  2
REC LOW-PAGE                  65  4 1

```

---

REC HIGH-PAGE	69	4	1	
REC MODE	73	2	1	
REC ACCESS	75	2	1	
REC \$ BFOR & AFTR RECORDS ONLY	*****			
REC IDMS-VSAM-FLAG	41	1	1	
REC AREA-TYPE	41	1	1	
REC VERB-NUM	42	1	1	
REC VERB	42	1	1	
REC PAGE-GROUP	43	2	1	
REC DBK-FORMAT	45	4	1	
REC DBK-LINES	48	1	1	
REC DB-KEY	49	4	1	
REC DB-KEY1	49	3	1	\$IDMS/1209
REC DB-KEY2	52	1	1	\$IDMS/1209
REC DB-KEY-A	59	4		\$For Hex compares.
REC USER-REC-ID	53	2	1	
REC USER-REC-ID-A	53	2		\$Hex compares IDMS/1893
REC PG-DISPL	55	2	1	
REC IMAGE-LEN	57	2	1	
REC PREFIX-LEN	59	2	1	
REC DISPLACEMENT	61	2	1	
REC SPAN-OFFSET	63	2	1	
REC USER-RECORD	65	2	1	

---

## 8.5 Summary of records required for journal reports

The table below lists the journal reports and journal records from which each report extracts data. Columns 3 through 10 are the journal record types, as follows:

- T — TIME
- B — BGIN
- A — AREA
- C — COMT
- E — ENDJ
- AB — ABRT
- BF — BFOR
- AF — AFTR

---

Report no. and title	T	B	A	C	E	AB	BF	AF
01 Transaction Summary		X		X	X	X		
02 Program Termination Statistics					X	X		
03 Program I/O Statistics					X	X		
04 Program Summary					X	X		
05 Detail Area/Transaction		X	X		X	X		
06 Detail Program/Area		X	X		X	X		
07 Area Summary		X	X		X	X		
08 Formatted Record Dump	X	X	X	X	X	X	X	X

---

## 8.6 Producing journal reports

### 8.6.1 Uses archive journal file as input

Journal reports are available at all sites where CA-CULPRIT and CA-IDMS/DB are installed. A journal report uses an archive journal file as input. The archive journal file is created by off-loading the disk journal file(s) to a tape or disk backup file using the ARCHIVE JOURNAL utility, as described in *CA-IDMS Utilities*. If the journal file is written directly to tape, the tape journal file can be used as input in place of the archive journal file. *Disk journal files that have not been offloaded cannot be used as the input file.*

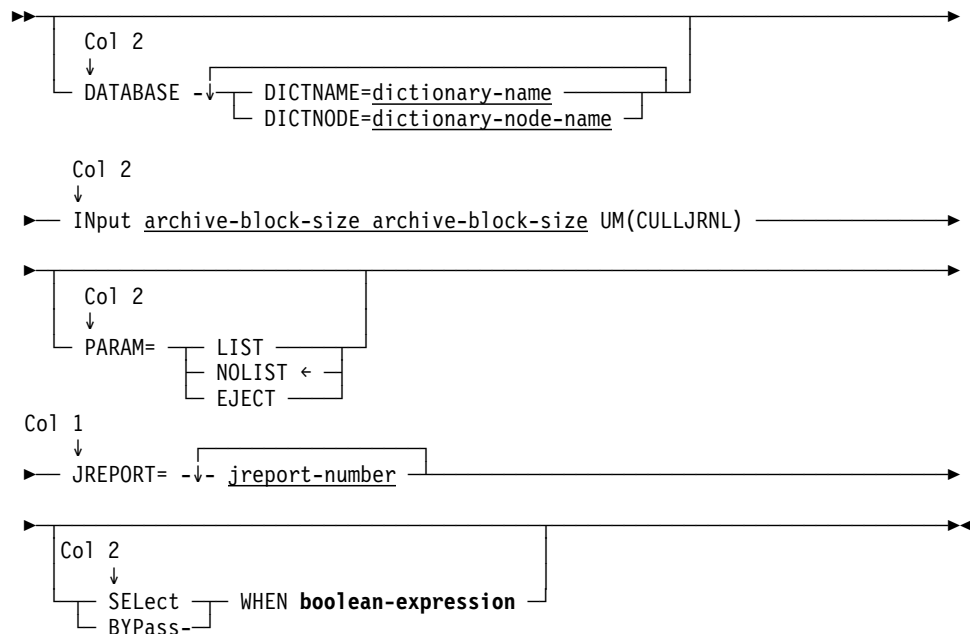
### 8.6.2 Required CA-CULPRIT modules

The following CA-CULPRIT modules, which are stored in the start-up dictionary at installation, are required to report on journal files:

- One input module (CULLJRNL) consisting of an Assembler routine that reads and deblocks the archive journal file and passes individual records to the CA-CULPRIT buffer
- One module (JREPORT 000) that provides REC parameters defining all fields contained in the journal records
- One of eight report modules (JREPORT 001 to JREPORT 008) that process and format information extracted from the journal file

### 8.6.3 Syntax

Syntax for the CA-CULPRIT input parameters is shown below, followed by examples. Coding is freeform, except for the starting column of the initial keyword in each parameter. JREPORT= begins in column 1; all other parameters begin in column 2.



### 8.6.4 Parameters

Syntax rules appear in Chapter 1, "Introduction," except as described below:

#### INPUT

Specifies the record size and block size of the input archive journal file and also names the CA-CULPRIT input module required to execute journal reports.

#### archive=block-size

Specifies the record size and block size of the archive journal file.

*Archive-block-size* must equal the archive journal block size specified in the physical DDL ARCHIVE JOURNAL statement.

**Note:** *Archive-block-size* must be specified twice.

#### UM(CULLJRNL)

Specifies the name of the CA-CULPRIT input module.

#### JREPORT=jreport-number

Identifies the reports to be printed. Multiple JREPORT statements can be submitted in a single run; each statement can specify one or more report numbers. The JREPORT statement must be coded starting in column 1.

*Jreport-number* specifies a journal report number in the range 001 through 008; leading zeros can be omitted. Multiple journal report numbers must be separated by blanks or commas.

#### SELECT/BYPASS WHEN **boolean-expression**

Specifies selection criteria to be applied to every journal record during the extract phase of processing.

The CA-CULPRIT buffer contains one journal record at a time. Selection criteria are applied to each record. Therefore, if the selection criteria pertain to a field in



one journal record type, additional selection criteria must be specified to select (rather than bypass) other journal record types required for a particular report.

**Note:** To determine which fields are contained in a journal record, refer to JREPORT-000. If a field is not in all journal record types, you must select the type in conjunction with the field to ensure correct record addressing. Field TYPE will be equal to one of the four-character journal record types documented under "Types of journal records."

## 8.6.5 Examples

**Example 1:** CA-CULPRIT retrieves JREPORT 002 and JREPORT 004 from the CULPDICT dictionary. The archived journal file contains 19068-byte, fixed-length records. By default, the CA-CULPRIT code will not be printed in the Sequential and Input Parameter Listings generated for the report.

```
DATABASE DICTNAME=CULPDICT
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=2,4
```

**Example 2:** In this example, selected records are printed in JREPORT 008 (Formatted Record Dump). The SELECT parameter selects only those BFOR and AFTR records that have a dbkey of X'000CEB08'.

```
DATABASE DICTNAME=CULPDICT
INPUT 4276 4276 UM(CULLJRNL)
JREPORT=008
SELECT TYPE EQ ('BFOR' 'AFTR') AND
*      DBKEY-A EQ (X'000CEB08')
```

**Example 3:** In this example, JREPORTs 005 and 006 are requested. By default, CA-CULPRIT retrieves the journal report modules from the system dictionary. For both reports, the CA-CULPRIT code will be printed on the Sequential Parameter Listing and on the Input Parameter Listing.

```
INPUT 19068 19068 UM(CULLJRNL)
PARAM=LIST
JREPORT=05
JREPORT=06
```

**Example 4:** This example code will print JREPORT 008 (Formatted Record Dump). The report will include only the BFOR and AFTR records within the specified sequence for run unit 44248. By default, the CULPRIT code for the report will not be printed in the parameter listings.

```
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=008
SEL WHEN TYPE EQ ('BFOR' 'AFTR')
*   AND TRANSACT-ID EQ 44248
*   AND SEQ EQ (1755732 TO 1755749)
```

**Example 5:** This example prints selected information for JREPORT 007, the Area Summary report. The SELECT parameter selects all AREA records that specify DDLDML or DDLDCLD. The SELECT parameter also selects record types BGIN, ENDJ, and ABRT, which are required for JREPORT 007.

```

INPUT 4276 4276 UM(CULLJRNL)
JREPORT=007
SELECT WHEN (TYPE EQ 'AREA' AND AREA EQ ('DDLML' DDLDCLOD'))
* OR TYPE EQ ('ABRT' 'BGIN' 'ENDJ')

```

### 8.6.6 Operating system considerations

Journal reports can run either in local mode or under central version. The JCL to run journal reports appears in Appendices A through D for OS/390, VSE/ESA, VM/ESA, and BS2000/OSD operating systems, respectively. The archived journal file must be defined with ddname/filename/linkname SYS010, as follows:

- **For OS/390 operating systems**, modify ddname SYS010 in Appendix A, as follows:

```
//SYS010 DD DSN=idms.archive,DISP=OLD,UNIT=tape,VOL=SER=nnnnnn
```

idms.archive	data set name of the archive journal file
tape	symbolic device name of the archive journal file
nnnnnn	volume serial number of the archive journal file

- **For VSE/ESA operating systems**, modify filename SYS010 in Appendix B, as follows:

```
//TLBL SYS010,'idms.archive'
//ASSGN SYS010,TAPE,VOL=nnnnnn
```

idms.archive	file-id of the archive journal file
nnnnnn	volume serial number

- **For VM/ESA operating systems**, modify ddname SYS010 in Appendix C, as follows:

```
For tape file: FILEDEF SYS010 TAP1 SL VOLID nnnnnn
                (RECFM FB LRECL 111 BLKSIZE bbbb
```

```
For disk file: FILEDEF SYS010 DISK input file a
```

nnnnnn	volume serial number of the archive journal file
lll	record length of the archive journal file
bbbb	block size of the archive journal file
input file a	filename, filetype, and filemode of the archive journal file

- **For BS2000/OSD operating systems**, modify linkname SYS010, as follows:

```
/ADD-FILE-LINK L-NAME=SYS010,F-NAME=idms.archive
```

idms.archive	filename of the archive journal file
--------------	--------------------------------------

---

## 8.7 JREPORT 001 — Transaction Summary

### 8.7.1 Purpose

JREPORT 001, the Transaction Summary report:

- Lists every program for which activity is recorded in the journal file
- Provides information on the duration and disposition of each transaction associated with the programs
- Extracts statistics from the BGIN, COMT, ENDJ, and ABRT journal records for each transaction.

The output is sorted by transaction identifier within program name.

The report below shows one page of a sample report:

REPORT NO. 01 JREPORT 001		IDMS JOURNAL REPORT R15.0 TRANSACTION SUMMARY										10/20/00	PAGE	1
PROGRAM NAME	NODE	TRANSACTION ID	TASK-ID	-- ORIGIN	ID --	----- DATE	START TIME	----- DATE	END TIME	TRM. REC.	COMMITTS			
DBCRUPD	SYSTEM72	0043	67	DBDC	67	03/02/00	04.33.52.88	03/02/00	04.33.52.93	ENDJ	0			
DBCRUPD	SYSTEM72	0046	111	DBDC	111	03/02/00	04.34.55.92	03/02/00	04.34.56.78	ENDJ	0			
DBCRUPD	SYSTEM72	0048	159	DBDC	159	03/02/00	04.34.56.95	03/02/00	04.34.56.97	ENDJ	0			
DBCRUPD	SYSTEM72	0051	186	DBDC	186	03/02/00	04.34.57.13	03/02/00	04.34.58.04	ENDJ	0			
DBCRUPD	SYSTEM72	0052	217	DBDC	217	03/02/00	04.34.57.14	03/02/00	04.34.58.04	ENDJ	0			
DBCRUPD	SYSTEM72	0055	233	DBDC	233	03/02/00	04.34.58.06	03/02/00	04.34.58.09	ENDJ	0			
DBCRUPD	SYSTEM72	0059	229	DBDC	229	03/02/00	04.34.58.08	03/02/00	04.34.58.09	ENDJ	0			
DBCRUPD	SYSTEM72	0058	230	DBDC	230	03/02/00	04.34.58.07	03/02/00	04.34.58.15	ENDJ	0			
DBCRUPD	SYSTEM72	0062	227	DBDC	227	03/02/00	04.34.58.10	03/02/00	04.34.58.15	ENDJ	0			
DBCRUPD	SYSTEM72	0064	225	DBDC	225	03/02/00	04.34.58.10	03/02/00	04.34.58.15	ENDJ	0			
DBCRUPD	SYSTEM72	0060	241	DBDC	241	03/02/00	04.34.58.14	03/02/00	04.34.58.16	ENDJ	0			
DBCRUPD	SYSTEM72	0057	231	DBDC	231	03/02/00	04.34.58.06	03/02/00	04.34.58.17	ENDJ	0			

Figure 8-1. Sample JREPORT 001

### 8.7.2 Field descriptions

A description of the fields in the Transaction Summary report follows:

#### PROGRAM NAME

Indicates the name of the program to which the information applies.

**Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND

RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

**NODE**

Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain \*local\*.

**TRANSACTION ID**

Specifies the unique identifier assigned to each transaction associated with the program.

**TASK**

Specifies the identifier of the task with which the transaction is associated. The task identifier is assigned by the DC/UCF system at run time.

**ORIGINATOR ID**

Specifies the identifier consisting of a 4-character name that designates the originating interface (for example, BATC, DBDC, or CICS) and a numeric identifier assigned to the transaction by the interface.

**START DATE/START TIME**

Specifies the date and time the transaction started.

**END DATE/END TIME**

Specifies the date and time the transaction ended.

**TERMINATION RECORDS**

Specifies the journal record type (ENDJ or ABRT) marking the termination of the transaction.

**COMMITTS**

Identifies the number of COMMIT records written for the transaction. (that is, the number of COMMIT or COMMIT WORK CONTINUE statements issued by the transaction).

## 8.8 JREPORT 002 — Program Termination Statistics

### 8.8.1 Purpose

JREPORT 002, the Program Termination Statistics report:

- Lists every program for which activity is recorded in the journal file.
- Contains detailed information on the database processing activities of each transaction associated with the programs.
- Extracts statistics from the ENDJ and ABRT journal records for each transaction.

The output is sorted by transaction identifier within program name.

The report below shows one page of a sample report.

REPORT NO. 02 JREPORT 002		IDMS JOURNAL REPORT R15.0 PROGRAM TERMINATION STATISTICS										10/20/00	PAGE	1
PROGRAM/ NODE	TRANSACTION ID	LOCKS REQUESTED	SHARED LOCKS HELD	EXCLUSIVE LOCKS HELD	-- CALC RECORDS -- STORED ON TARGET	NOT STORED ON TARGET	-- VIA RECORDS -- STORED ON TARGET	NOT STORED ON TARGET	---- RECORDS ---- REQUESTED FROM DB	CURRENT OF TRANS	CALLS TO DB	--FRAGMENTS-- STORED RETND		
DBCRUPD SYSTEM72	0043	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0044	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0045	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0046	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0047	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0048	21	6	7			1		11	4	16			
DBCRUPD SYSTEM72	0050	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0051	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0052	21	6	7			1		11	4	16			
DBCRUPD SYSTEM72	** ABORT ** 0053	12	4	5			1		6	3	11			
DBCRUPD SYSTEM72	0054	17	6	7			1		11	4	15			
DBCRUPD SYSTEM72	0055	17	6	7			1		11	4	15			

Figure 8-2. Sample JREPORT 002

## 8.8.2 Field descriptions

A description of the fields in the Program Termination Statistics report follows:

### **PROGRAM/NODE**

Identifies the name of the program to which the information applies and the node name on which it executed. Transactions that terminate abnormally are flagged as ABORT.

**Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

### **TRANSACTION ID**

Specifies the unique identifier assigned to each transaction.

### **LOCKS REQUESTED**

Indicates the number of locks acquired by the transaction.

### **SHARED LOCKS HELD**

Indicates the number of shared locks held by the transaction at the time the transaction ended.

### **EXCLUSIVE LOCKS HELD**

Indicates the number of exclusive locks held by the transaction at the time the transaction ended.

### **CALC RECORDS STORED ON TARGET/NOT STORED ON TARGET**

Indicates the number of CALC records stored/not stored on their target page.

Interpretation: The ratio of CALC records stored on target to the total number stored (that is, hits plus overflows) is the **CALC cluster ratio**. The ratio reflects the efficiency of the CALC algorithm. Ideally, the value should be 1, which indicates no overflow. Values less than 1 or less than the norm indicate space utilization is getting high and database tuning should be performed.

### **VIA RECORDS STORED ON TARGET/NOT STORED ON TARGET**

Identifies the number of VIA and/or DIRECT records stored/not stored on their target page.

Interpretation: The ratio of VIA records stored on target to the total number stored (that is, hits plus overflows) is the **VIA cluster ratio**. The ratio reflects how well VIA records cluster around their owner. Ideally, the value should be 1, which indicates no overflow. Values less than 1 or less than the norm indicate very large data clusters, high utilization of space, or small page size.

### **RECORDS REQUESTED FROM DB**

Indicates the number of database records requested by IDMS/DBMS.

### **RECORDS CURRENT OF TRANS**

Indicates the number of times the current-of-transaction field in the subschema control block for the transaction was updated.

Interpretation: The ratio of records requested to records current-of-transaction is the **effectiveness ratio**. The ratio measures the amount of work CA-IDMS/DB is doing for the programmer (that is, how many records the DBMS has to examine to find the one requested). The information in JREPORT 002 indicates whether the ratio is constant for all executions of the program or only for certain transactions.

The value should be as low as possible. If the value is high, examine set options (for example, sorted order) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

### **CALLS TO DB**

Indicates the number of calls to IDMSDBMS issued by the transaction. Execution of each navigational DML request involves one call; execution of each LRF and SQL request typically involves multiple calls.

### **FRAGMENTS STORED**

Indicates the number of noncontiguous segments (fragments) stored for variable-length records.

Interpretation: If the number of stored fragments is large or increasing, increase the page size to accommodate larger records. The condition of the databases and areas involved with the program should be analyzed further with the Database Analysis Utility (IDMSDBAN), described in Chapter 11, "Other CA-IDMS Reporting Facilities."

### **FRAGMENTS RETND**

Indicates the number of records relocated from their home page.



## 8.9 JREPORT 003 — Program I/O Statistics

### 8.9.1 Purpose

JREPORT 003, the Program I/O Statistics report:

- Lists every program for which activity is recorded in the journal file
- Provides information on the I/O services requested by and performed for each transaction associated with the programs
- Extracts statistics from the ENDJ and ABRT journal records for each transaction.

The output is sorted by transaction identifier within program name.

The report below shows one page of a sample report:

REPORT NO. 03 JREPORT 003			IDMS JOURNAL REPORT PROGRAM I/O STATISTICS ----- PAGES -----			R15.0	10/20/00	PAGE	1
PROGRAM	NODE	TRANSACTION ID	READ	WRITTEN	REQUESTED				
DBCRUPD	SYSTEM72	0043	2	2	7				
DBCRUPD	SYSTEM72	0044	2	2	7				
DBCRUPD	SYSTEM72	0045	2	2	7				
DBCRUPD	SYSTEM72	0046	2	2	7				
DBCRUPD	SYSTEM72	0047	2	2	7				
DBCRUPD	SYSTEM72	0048	2	2	7				
DBCRUPD	SYSTEM72	0050	2	2	7				
DBCRUPD	SYSTEM72	0051	2	2	7				
DBCRUPD	SYSTEM72	0052	2	2	7				
DBCRUPD	SYSTEM72	0053	2	1	4				
DBCRUPD	SYSTEM72	0054	2	2	7				
DBCRUPD	SYSTEM72	0055	2	2	7				

Figure 8-3. Sample JREPORT 003

### 8.9.2 Field descriptions

A description of the fields in the Program I/O Statistics report follows:

#### PROGRAM

Identifies the name of the program to which the information applies.

**Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

#### NODE

Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain \*local\*.

**TRANSACTION ID**

Is the unique identifier assigned to each transaction associated with the program. Transactions that terminate abnormally are flagged as ABORT.

**PAGES READ**

Indicates the number of pages physically read from disk for the transaction.

**PAGES WRITTEN**

Indicates the number of pages physically written to disk for the transaction. A page can be updated several times before it is actually written back to the database.

**PAGES REQUESTED**

Indicates the number of pages requested by IDMSDBMS (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool.

Interpretation: The ratio of pages requested to pages read is the **buffer utilization ratio**. It measures the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio the better. Ratios consistently below 2.0 indicate that processing is random or that the buffer-pool size is too small. The information in JREPORT 003 indicates whether the ratio is constant for all executions of the program or only for certain transactions.

**Note:** The buffer utilization ratio may be artificially high for transactions that keep locks. IDMSDBMS cannot hold a buffer while waiting for a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a lock is requested for which a wait must occur.

## 8.10 JREPORT 004 — Program Summary

### 8.10.1 Purpose

JREPORT 004, the Program Summary report:

- Lists every program for which activity is recorded in the journal file
- Provides summary information on the database processing activities of the program
- Compiles statistics from information extracted from the ENDJ and ABRT journal records for transactions associated with each program
- Indicates the averages and ratios for all programs combined

The output is sorted by program name.

The report below shows one page of a sample report:

REPORT NO. 04 JREPORT 004		IDMS JOURNAL REPORT PROGRAM SUMMARY						R15.0	10/20/00	PAGE	1
		----- AVERAGES -----				----- RATIOS -----					
PROGRAM NAME	TIMES RUN	PAGES READ	PAGES WRITTEN	CALLS	RECORDS REQUESTED	BUFFER UTILIZATION	EFFECTIVENESS	CALC CLUSTER	SPACE MANAGEMENT	VIA CLUSTER	
DBCRUPD	881	2	2	15	12	4.03	2.99		7.04	1.00	
TOTAL	881	2	2	15	12	4.03	2.99		7.04	1.00	

Figure 8-4. Sample JREPORT 004

### 8.10.2 Field descriptions

A description of the fields in the Program Summary report follows:

#### PROGRAM NAME

Identifies the name of the program to which the information applies.

**Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

#### TIMES RUN

Indicates the number of times the program was run.

#### AVERAGES/ PAGES READ

Indicates the average number of pages physically read from disk for each run of the program.

#### **AVERAGES/ PAGES WRITTEN**

Indicates the average number of pages physically written to disk for each run of the program.

#### **AVERAGES/ CALLS**

Indicates the average number of calls to IDMSDBMS issued for each run of the program.

#### **AVERAGES/ RECORDS REQUESTED**

Indicates the average number of database records requested by IDMSDBMS for each run of the program.

#### **RATIOS/ BUFFER UTILIZATION**

Indicates the ratio of number of pages requested to pages read.

Interpretation: This ratio indicates the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio, the better. Ratios consistently below 2.0 indicate that processing may be random or that the buffer-pool size may be too small.

Particular attention should be given to frequently used programs. If the ratio for a program is below 2.0, users can look at the statistics generated by JREPORT 003 to determine whether the number of pages requested and pages read were constant for all executions of the program or only for certain transactions.

**Note:** The buffer utilization ratio may be artificially high for transactions that keep locks. IDMSDBMS cannot hold a buffer while waiting for a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a lock is requested for which a wait must occur.

#### **RATIOS/ EFFECTIVENESS**

Indicates the ratio of records requested to records made current-of-run-unit.

Interpretation: The effectiveness ratio indicates the amount of work CA-IDMS/DB is doing for the programmer (that is, how many records IDMSDBMS has to examine to find the one requested). The lower the ratio the better. If the ratio is very high, examine set options (for example, sort order or next pointers only) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

#### **RATIOS/ CALC CLUSTER**

Indicates the ratio of the number of CALC records stored on their target page to the total stored (that is, hits plus overflows). The ratio reflects the efficiency of the CALC algorithm.

Interpretation: The CALC cluster ratio is especially important when the database is loaded or restructured. Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate that space utilization is getting high and database tuning should be performed.

#### **RATIOS/ SPACE MANAGEMENT**

Indicates the ratio of records requested by IDMSDBMS to pages read from the database.

Interpretation: The space management ratio measures how well space is allocated (for example, VIA options, CALC distribution, and buffering). The higher the ratio the better. Ratios less than 4 or less than the norm indicate that the size of the buffer should be increased and database tuning should be performed.

**Note:** The space management ratio may be artificially high for transactions that keep locks. IDMSDBMS cannot hold a buffer while waiting for a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a lock is requested for which a wait must occur.

#### **RATIOS/ VIA CLUSTER**

Specifies the ratio of VIA records stored on the same page as their owner to the total number of VIA records stored (that is, hits plus overflows). The ratio reflects how well VIA records cluster around their owner.

Interpretation: The VIA cluster ratio gives an indication of VIA storage requirements and is an indirect measure of the amount of space still available in the area. Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate very large data clusters, high utilization of space, or small page size. The cause may be a database design in which there are too many VIA records (of a particular record type) or in which the VIA records are too large.

## 8.11 JREPORT 005 — Detail Area/Transaction

### 8.11.1 Purpose

JREPORT 005, the Detail Area/Transaction report:

- Lists every database area identified in the journal file
- Provides detailed information on transaction access to the area
- Extracts information from the BGIN, AREA, ENDJ, and ABRT journal records for each transaction

The output is sorted by transaction identifier within program name within area name.

The report below shows one page of a sample Detail Area/Transaction report:

REPORT NO. 05 JREPORT 005		IDMS JOURNAL REPORT R15.0 DETAIL AREA/TRANSACTION		10/20/00	PAGE 1
AREA NAME	PROGRAM NAME	NODE	TRANSACTION ID	OPEN ACCESS	OPEN MODE
ACCTHIST	DBCRUPD	SYSTEM72	0043	SHARED	UPDATE
		SYSTEM72	0044	SHARED	UPDATE
		SYSTEM72	0045	SHARED	UPDATE
		SYSTEM72	0046	SHARED	UPDATE
		SYSTEM72	0047	SHARED	UPDATE
		SYSTEM72	0048	SHARED	UPDATE
		SYSTEM72	0050	SHARED	UPDATE
		SYSTEM72	0051	SHARED	UPDATE
		SYSTEM72	0052	SHARED	UPDATE
		SYSTEM72	0053	SHARED	UPDATE
		SYSTEM72	0054	SHARED	UPDATE
		SYSTEM72	0055	SHARED	UPDATE
BRNCHTEL	DBCRUPD	SYSTEM72	0043	SHARED	UPDATE
		SYSTEM72	0044	SHARED	UPDATE
		SYSTEM72	0045	SHARED	UPDATE
		SYSTEM72	0046	SHARED	UPDATE
		SYSTEM72	0047	SHARED	UPDATE
		SYSTEM72	0048	SHARED	UPDATE
		SYSTEM72	0050	SHARED	UPDATE
		SYSTEM72	0051	SHARED	UPDATE
		SYSTEM72	0052	SHARED	UPDATE
		SYSTEM72	0053	SHARED	UPDATE
		SYSTEM72	0054	SHARED	UPDATE
		SYSTEM72	0055	SHARED	UPDATE

Figure 8-5. Sample JREPORT 005

### 8.11.2 Field descriptions

A description of the fields in the Detail Area/Transaction report follows:

**AREA NAME**

Specifies the subschema name of the area to which the access information applies.

**PROGRAM NAME**

Specifies the name of each program that accesses the area.

**Note:** The subschema name field supplies the name of the program. If the

program name is blank, then the program issued a BIND RUN UNIT before moving its name to the subschema name field.

**NODE**

Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain \*local\*.

**TRANSACTION ID**

Identifies the unique identifier assigned to each transaction associated with the program.

**OPEN ACCESS**

Identifies area usage mode options that prevent update or retrieval of an area by other transaction executing concurrently under the central version. Possible options are SHARED (default), PROTECTED, and EXCLUSIVE.

**OPEN MODE**

Identifies the mode in which the transaction opened the area. Possible modes are:

- UPDATE — The transaction can issue all DML functions for records in the area
- RETRIEVAL — The transaction cannot issue DML or DDL requests that result in updates to data in the area

## 8.12 JREPORT 006 — Detail Program/Area

### 8.12.1 Purpose

JREPORT 006, the Detail Program/Area report:

- Lists every program for which activity is recorded in the journal file
- Provides detailed information on the areas accessed by each transaction associated with the programs
- Extracts information from the BGIN, AREA, ENDJ, and ABRT journal records for each transaction

With this report, you can identify what areas are used exclusively by certain programs and are therefore not available to concurrently running programs. The output is sorted by area name within transaction identifier within program name.

The report below shows one page of a sample Detail Program/Area report:

REPORT NO. 06 JREPORT 006		IDMS JOURNAL REPORT DETAIL PROGRAM/AREA		R15.0	10/20/00	PAGE	1
PROGRAM NAME	NODE	TRANSACTION ID	AREA NAME	OPEN ACCESS	OPEN MODE		
DBCRUPD	SYSTEM72	0043	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0043	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0044	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0044	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0045	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0045	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0046	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0046	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0047	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0047	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0048	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0048	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0050	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0050	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0051	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0051	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0052	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0052	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0053	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0053	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0054	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0054	BRNCHTEL	SHARED	UPDATE		
	SYSTEM72	0055	ACCTHIST	SHARED	UPDATE		
	SYSTEM72	0055	BRNCHTEL	SHARED	UPDATE		

Figure 8-6. Sample JREPORT 006

### 8.12.2 Field descriptions

A description of the fields in the Detail Program/Area report follows:

#### PROGRAM NAME

Identifies the name of the program to which the information applies.

**Note:** The subschema name field supplies the name of the program. If the



program name is blank, then the program issued a BIND RUN UNIT before moving its name to the subschema name field.

**NODE**

Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain \*local\*.

**TRANSACTION ID**

Identifies the unique identifier assigned to each transaction associated with the program.

**AREA NAME**

Identifies the subschema name of each area accessed by the transaction.

**OPEN ACCESS**

Identifies the area usage mode options that prevent update or retrieval of an area by other transactions executing concurrently under the central version. Possible options are SHARED (default), PROTECTED, and EXCLUSIVE.

**OPEN MODE**

Identifies the mode in which the transaction opened the area. Possible modes are:

- UPDATE — The transaction can issue all DML functions for records in the area
- RETRIEVAL — The transaction cannot issue DML or DDL requests that result in updates to data in the area

## 8.13 JREPORT 007 — Area Summary

### 8.13.1 Purpose

JREPORT 007, the Area Summary report:

- Lists every area identified in the journal file
- Provides summary information on program access to the area
- Extracts statistics from the BGIN, AREA, ENDJ, and ABRT journal records for each transaction
- Indicates the total number of times all areas were readied in each mode.

The output is sorted by program name within area name.

The report below shows one page of a sample Area Summary report:

REPORT NO. 07 JREPORT 007		IDMS JOURNAL REPORT AREA SUMMARY			R15.0	10/20/00			PAGE	1
AREA NAME	PROGRAM NAME	----- RETRIEVAL -----				----- UPDATE -----				
		SHARED	EXCLUSIVE	PROTECTED		SHARED	EXCLUSIVE	PROTECTED		
ACCTHIST	DBCRUPD					881				
BRNCHTEL	DBCRUPD					881				
TOTAL FOR ALL AREAS						1762				

Figure 8-7. Sample JREPORT 007

### 8.13.2 Field descriptions

A description of the fields in the Area Summary report follows:

#### AREA NAME

Identifies the subschema name of the area to which the information applies.

#### PROGRAM NAME

Identifies the name of each program that accesses the area.

**Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

#### SHARED RETRIEVAL

Indicates the number of times transactions associated with the program readied the area in shared retrieval mode.

**EXCLUSIVE RETRIEVAL**

Indicates the number of times transactions associated with the program readied the area in exclusive retrieval mode.

**PROTECTED RETRIEVAL**

Indicates the number of times transactions associated with the program readied the area in protected retrieval mode.

**SHARED UPDATE**

Indicates the number of times transactions associated with the program readied the area in shared update mode.

**EXCLUSIVE UPDATE**

Indicates the number of times transactions associated with the program readied the area in exclusive update mode.

**PROTECTED UPDATE**

Indicates the number of times transactions associated with the program readied the area in protected update mode.

## 8.14 JREPORT 008 — Formatted Record Dump

### 8.14.1 Purpose

JREPORT 008, the Formatted Record Dump:

- Produces a formatted dump of TIME, BGIN, AREA, COMT, ENDJ, ABRT, BFOR, and AFTR journal records, allowing the user to inspect the entire contents of the journal file.
- Provides a character and hexadecimal dump of the user record portion of BFOR and AFTR journal entries.
- Includes JSEG (journal segment) dumps in character and hexadecimal format. For the layout of the JSEG journal header record, see DSECT #JTRDS in the *CA-IDMS DSECT Reference*.

The report below shows one page from a sample Formatted Record Dump:

### 8.14.2 Field descriptions

## JOURNAL RECORD

Chapter 8. CA-IDMS/DB Journal Reports — JREPORTS 8-37

**NODE NAME**

Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain \*local\*.

**UTC DATE**

Identifies the UTC date on which the journal record was either written to the journal file (TIME record only) or placed into the journal buffer (all other records). The date is given in the same form as LOCAL DATE.

**UTC TIME**

Identifies the UTC time at which the journal record was either written to the journal file (TIME record only) or placed into the journal buffer (all other records). The date is given in the same form as LOCAL TIME.

**JOURNAL SEQUENCE**

Identifies the sequence number assigned to the journal record.

**TRANSACT ID**

Identifies the unique identifier assigned to the transaction for which the journal record was written.

**VERB NUM.**

(BFOR and AFTR records only) Identifies the function code of the navigational DML verb issued by an application program or the SQL or LRF runtime processor.

**PG GRP**

(AREA, BFOR and AFTR records only) Identifies the page group associated with the area in which the user record is stored.

**PROGRAM NAME**

(BGIN, COMT, ENDJ, and ABRT records only) Identifies the name of the program with which the transaction is associated.

**LOCAL DATE**

Identifies the date on which the journal record was either written to the journal file (TIME record only) or placed into the journal buffer (BGIN, COMT, ENDJ, and ABRT records only). The date is given in the form *mm/dd/yy*, where *mm* is the month, *dd* is the day, and *yy* is the last two digits of the year.

**LOCAL TIME**

Identifies the time at which the journal record was either written to the journal file (TIME record only) or placed into the journal buffer (BGIN, COMT, ENDJ, and ABRT records only). The time is given in the form *hhmmssht*, where *hh* is hours on a 24-hour clock, *mm* is minutes, *ss* is seconds, and *ht* is hundredths of a second.

**TRANSACT QUIESCE LEVEL**

(BGIN, COMT, ENDJ, and ABRT records only) Identifies the number of open transactions after the journal record was written to the journal file.

**LOCAL TASK ID**

(BGIN, COMT, ENDJ, and ABRT records only) Is the identifier consisting of a 4-character name that designates the originating interface (for example, BATC,

DBDC, or CICS) and a numeric identifier assigned to the transaction by the interface.

**CALLS TO DB**

(COMT, ENDJ, and ABRT records only) Indicates the number of calls to IDMSDBMS issued by the transaction. Execution of each navigational DML request involves one call; execution of each LRF or SQL request typically involves multiple calls. On COMT checkpoints, this value is a running total for the transaction.

**PAGES REQUESTED**

(COMT, ENDJ, and ABRT records only) Indicates the number of pages requested by the transaction (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool. On COMT checkpoints, this value is a running total for the transaction.

**PAGES READ**

(COMT, ENDJ, and ABRT records only) Indicates the number of pages physically read from disk. On COMT checkpoints, this value is a running total for the transaction.

**PAGES WRITTEN**

(COMT, ENDJ, and ABRT records only) Indicates the number of pages physically written to disk. On COMT checkpoints, this value is a running total for the transaction.

**RECDs REQUESTED**

(COMT, ENDJ, and ABRT records only) Indicates the number of database records requested by IDMSDBMS for the transaction. On COMT checkpoints, this value is a running total for the transaction.

**RECORDS UPDATED**

(COMT, ENDJ, and ABRT records only) Indicates the number of records updated by the transaction. On COMT checkpoints, this value is a running total for the transaction.

**CURRENT OF TR**

(COMT, ENDJ, and ABRT records only) Indicates the number of times the current-of-transaction field in the subschema control block for the transaction was updated. On COMT checkpoints, this value is a running total for the transaction.

**CALC ON TARGET**

(COMT, ENDJ, and ABRT records only) Identifies the number of CALC records stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

**CALC NOT TARGET**

(COMT, ENDJ, and ABRT records only) Identifies the number of CALC records not stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

**VIA ON TARGET**

(COMT, ENDJ, and ABRT records only) Identifies the number of VIA and/or DIRECT records stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

**VIA NOT TARGET**

(COMT, ENDJ, and ABRT records only) Identifies the number of VIA and/or DIRECT records not stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

**FRAGMNTS STORED**

(COMT, ENDJ, and ABRT records only) Identifies the number of noncontiguous segments stored for variable-length records. On COMT checkpoints, this value is a running total for the transaction.

**RECDs RELOCATED**

(COMT, ENDJ, and ABRT records only) Identifies the number of records relocated from their home page. On COMT checkpoints, this value is a running total for the transaction.

**LOCKS REQUESTED**

(COMT, ENDJ, and ABRT records only) Indicates the number of locks acquired by the transaction. On COMT checkpoints, this value is a running total for the transaction.

**SHARED LOCKS**

(COMT, ENDJ, and ABRT records only) Indicates the number of shared locks held by the transaction at the time the checkpoint record was written.

**EXCLUSIVE LOCKS**

(COMT, ENDJ, and ABRT records only) Indicates the number of exclusive locks held by the transaction at the time the checkpoint record was written.

**Area name**

(AREA records only) Identifies the name of the area for which the checkpoint record was written.

**LOW PAGE**

(AREA records only) Identifies the page number of the first page in the area.

**HIGH PAGE**

(AREA records only) Identifies the page number of the last page in the area.

**Open access**

(AREA records only) Identifies the mode in which the transaction accessed the area (SHARED, PROTECTED, or EXCLUSIVE).

**Open mode**

(AREA records only) Identifies the mode in which the transaction opened the area (RETRIEVAL or UPDATE).

**THIS IS A NATIVE IDMS RECORD**

(BFOR and AFTR records only) Indicates that the user record is a CA-IDMS/DB database record. For native VSAM records, this field reads THIS IS A NATIVE VSAM RECORD.



**PG DISPLACEMENT**

(BFOR and AFTR records only) Specifies the location of the user record occurrence relative to the beginning of the database page (given as a decimal offset).

**PREFIX LENGTH**

(BFOR and AFTR records only) Specifies the length in bytes of the prefix portion of the user record.

**USER RECORD ID**

(BFOR and AFTR records only) Indicates the record id of the user record. An asterisk following the record id indicates that this is a logically deleted record.

**USER RECD LGTH**

(BFOR and AFTR records only) Specifies the length in bytes of the entire user record (prefix and data portion) as found in the database record line index.

**VERB NUMBER**

(BFOR and AFTR records only) Specifies the function code of the navigational DML verb issued by an application program or the SQL or LRF runtime processor.

**USER RECORD DB KEY**

(BFOR and AFTR records only) Specifies the database key of the user record occurrence in hexadecimal format.

**PAGE/LINE**

(BFOR and AFTR records only) Specifies the page and line number, in decimal format, of the database key of the user record occurrence.

**DISPLACEMENT OF CALL**

(BFOR and AFTR records only) Specifies a trace entry indicating the IDMSDBMS routine that issued the database call (for CA internal use only).

**CHAR**

Identifies the contents of the user record in decimal (display) format.

**ZONE NUMR**

Identifies the contents of the user record in hexadecimal format.

**SPANNED OFFSET**

(Not on sample report) Indicates that the journal entry is a continuation of the previous BFOR or AFTR image for the transaction. The number is the displacement of this portion of the record image relative to the entire image.



## Chapter 9. DC/UCF Statistics Reports — SREPORTS

---

9.1 Overview	9-3
9.2 Summary of statistics reports	9-4
9.3 Uses for statistics reports	9-5
9.4 Other tools available	9-6
9.5 Types of statistics records	9-7
9.5.1 Statistics record type	9-7
9.5.2 Layout of statistics log records	9-7
9.5.3 Task or transaction statistics	9-8
9.5.4 Order of DSECTs for task and transaction statistics	9-8
9.5.5 System statistics	9-9
9.5.6 Order of DSECTs for system statistics	9-9
9.5.7 Histograms	9-9
9.5.8 Order of DSECTs for histograms	9-9
9.5.9 Release level	9-10
9.5.10 Number of statistics log records	9-10
9.5.11 Output of SREPORT 099	9-10
9.5.12 SREPORT 000	9-10
9.6 Producing statistics reports	9-11
9.6.1 Syntax	9-11
9.6.2 Parameters	9-12
9.6.3 Examples	9-13
9.6.4 Operating system considerations	9-14
9.7 DC/UCF system statistics reports	9-17
9.7.1 SREPORT 003 — IDMS-DC System Statistics	9-17
9.7.2 SREPORT 012 — IDMS-DC Task Summary	9-27
9.7.3 SREPORT 013 — IDMS-DC Program Summary	9-29
9.7.4 SREPORT 014 — IDMS-DC Queue Summary	9-31
9.7.5 SREPORT 015 — IDMS-DC Line Summary	9-31
9.7.6 SREPORT 016 — IDMS-DC Physical Terminal Summary	9-33
9.8 Task and External Request Unit Service (ERUS) statistics reports	9-35
9.8.1 Field descriptions	9-38
9.9 Transaction statistics reports	9-43
9.9.1 Field descriptions	9-45
9.10 CA-ADS dialog statistics	9-46
9.10.1 Field descriptions	9-49
9.11 Histogram Report	9-53
9.11.1 Field descriptions	9-55
9.12 Record Summary Statistics Report	9-57
9.12.1 Contents of SREPORT 017	9-58
9.12.2 Field descriptions	9-59



## 9.1 Overview

CA-IDMS/DC and CA-IDMS/UCF (DC/UCF) collect the following run-time statistics:

- **System statistics**, which record resource usage for an entire system
- **Task and external request unit service (ERUS) statistics**, which record resource usage by task
- **Transaction statistics**, which record resource usage by transaction
- **CA-ADS dialog statistics**, which record dialog activity associated with an application
- **Histogram statistics**, which record resource usage by frequency of occurrence within a value range
- **Record statistics**, which document the types of statistics records in the archived log file

## 9.2 Summary of statistics reports

The table below lists the reports that document these statistics in order by report module number:

<b>SREPORT Module</b>	<b>SREPORT Category</b>	<b>Statistics Report Title</b>
000		Startup Records Read (required with remaining modules)
001	Histogram	IDMS Statistics Histogram Report (system and task)
003	System	IDMS-DC System Statistics
005	Task	IDMS-DC Task Statistics by User Id
006	Task	IDMS-DC Task Statistics by Lterm Id
007	Task	IDMS-DC Task Statistics by Task Code
008	Task	IDMS-DC ERUS Task Statistics by Accounting Data
009	Task	IDMS-DC ERUS Task Statistics by Program Name
010	Transaction	IDMS-DC Transaction Statistics by User Id
011	Transaction	IDMS-DC Transaction Statistics by Lterm Id
012	System	IDMS-DC Task Summary
013	System	IDMS-DC Program Summary
014	System	IDMS-DC Queue Summary
015	System	IDMS-DC Line Summary
016	System	IDMS-DC Physical Terminal Summary
017	Record	Summary of Records Read
018	CA-ADS	ADS/OnLine Statistics by User Id
019	CA-ADS	ADS/OnLine Statistics by Dialog and Version Number
020	CA-ADS	ADS/OnLine Statistics by Logical Terminal Id
021	Transaction	IDMS-DC Transaction Statistics by Dialog
099		No listing (creates an output file of archive statistics records)

## 9.3 Uses for statistics reports

Systems administrators use statistics reports to monitor system activity. Statistics reports can be used to:

- Assist in system tuning and maintenance
- Evaluate processing efficiency at the system level, task level, transaction level, and dialog level
- Monitor system activity within certain time intervals

## 9.4 Other tools available

Other tools available to monitor system activity are the PRINT LOG utility, PLOG, and DCMT STATISTICS commands. These tools are discussed in more detail in Chapter 11, "Other CA-IDMS Reporting Facilities."

**What follows:** This chapter discusses types of statistics records, input parameters needed to process a statistics report, and sample output.

►► For more information about statistics, see *CA-IDMS System Operations*.



## 9.5 Types of statistics records

### 9.5.1 Statistics record type

DC/UCF logs six types of records to the system log file. One of these record types is a statistics record. A statistics record is variable in length and has two parts. The first part contains information such as the date and time. The second part contains statistical data for the 35 types of statistics records. Three DSECTS (documented in the *CA-IDMS DSECT Reference*) define the layout of the statistics records, as follows:

DSECT	Description
#LGRDS	The log record header DSECT
#STLDS	The statistics log record header DSECT. DSECT #STLDS redefines 256 bytes of #LGRDS starting at offset X '14' and contains a description of 35 statistics record subtypes.
#STRDS	The statistics record DSECT (#STRDS). #STRDS describes the layout of the statistics in the log record.
#HSTDS	The histogram record DSECT (#HSTDS). #HSTDS describes the layout of histogram data in the log record. Histogram log records are variable length. The DSECT redefines 244 bytes of #STLDS starting at offset X '0C'.

►► Each of these DSECTS is presented in the *CA-IDMS DSECT Reference*.

### 9.5.2 Layout of statistics log records

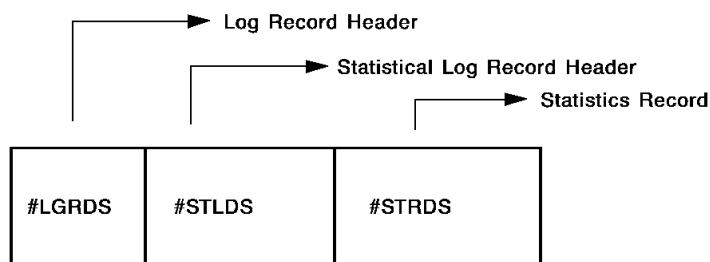
The layout of statistics log records varies depending on the type of statistics being saved. For all statistics log records, the first two DSECTS are:

- #LGRDS
- #STLDS

These two DSECTS are followed by one or more occurrences of #STRDS and, for histogram log records, by #HSTDS.

The figure below illustrates the order of DSECTS for all statistics log records *except*:

- Task and transaction statistics
- System statistics
- Histograms



For a list of statistics log records, see Table 9-1 on page 9-53, later in this chapter.

### 9.5.3 Task or transaction statistics

For task or transaction statistics, there are three #STRDS DSECTs:

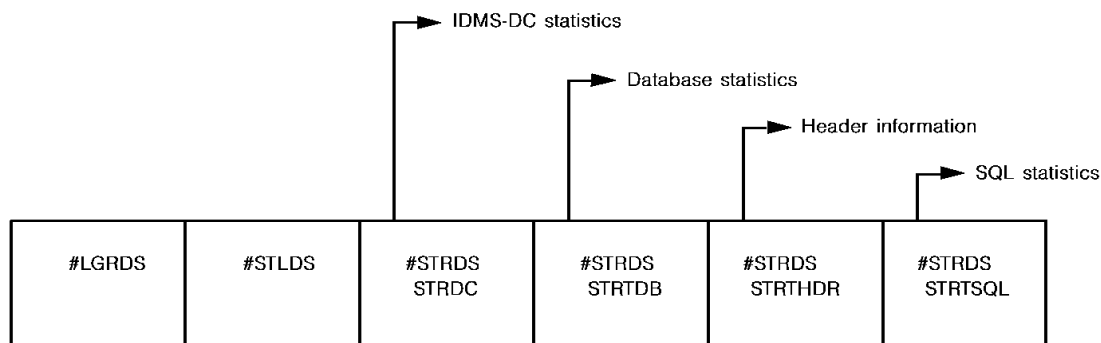
- The first DSECT describes the layout of DC statistics.
- The second DSECT describes the layout of database statistics.
- The third DSECT describes header information.
- The fourth DSECT describes the layout of SQL statistics.

To determine whether the current log record is for task or transaction statistics, check the STLTYPE field in #STLDS:

Transaction statistics	STLTYPE=STLTTSB
Task statistics	STLTYPE=STLTTST

### 9.5.4 Order of DSECTs for task and transaction statistics

The figure below illustrates the order of DSECTs for task and transaction statistics log records:



### 9.5.5 System statistics

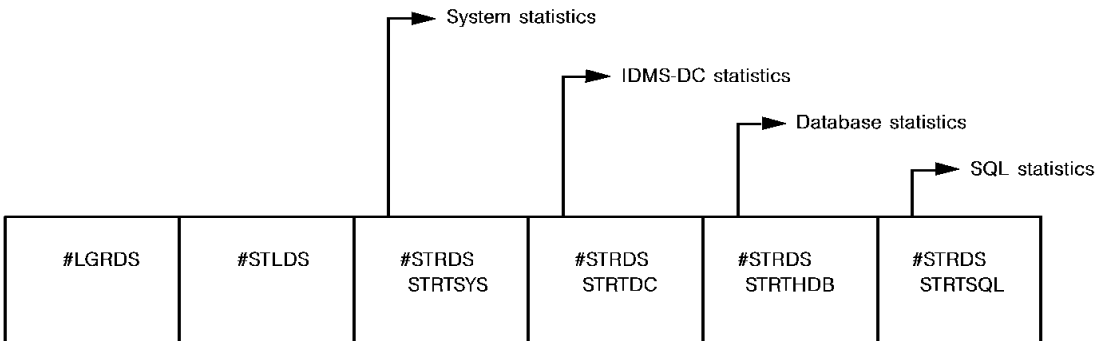
For system statistics, there are three #STRDS DSECTs:

- The first DSECT describes the layout of system statistics.
- The second DSECT describes the layout of IDMS-DC statistics.
- The third DSECT describes the layout of database statistics.
- The fourth DSECT describes the layout of SQL statistics.

To determine whether the current log record is for system statistics, check the STLTYPE field in #STLDS. For system statistics, STLTYPE=STLTCST.

### 9.5.6 Order of DSECTs for system statistics

The figure below illustrates the order of DSECTs for system statistics log records:



### 9.5.7 Histograms

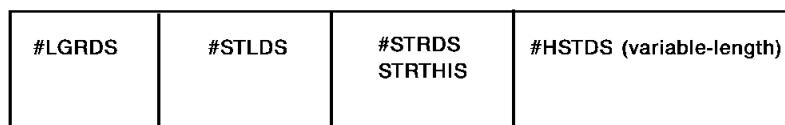
For histograms, there is one #STRDS DSECT followed by one #HSTDS DSECT.

To determine whether the current log record is for a histogram:

1. Check the STLTYPE field in #STLDS. It should be either STLTCST or STLTPL.
2. Check the STRTYPE field in #STRDS. It should be STRTHIS.

### 9.5.8 Order of DSECTs for histograms

The figure below illustrates the order of DSECTs for histogram log records:



## 9.5.9 Release level

The #STRDS DSECT contains a field that indicates whether the statistics log record is for a Release 15.0 DC/UCF system.

To determine whether the record is for Release 15.0, check the STRRID field in #STRDS. For Release 15.0, STRRID=C'R150'.

## 9.5.10 Number of statistics log records

A statistics block can contain more data than can be written in one log record. When this occurs, DC/UCF uses two log records to write the statistics to the log.

To determine whether a particular statistics log record contains only part of the data from a statistics block, check the LGRTYPE field in #LGRDS:

- If LGRTYPE=X'76', this is the first of two log records used to write one statistics block.
- If LGRTYPE=X'F6', this log record is one of the following:
  - The only log record used for that particular statistics block
  - The second of two log records used to write one statistics block

**Note:** Statistics log records have a maximum length of 276 bytes. Programs that read statistics from the DC/UCF log should reserve twice that amount of space to handle statistics that span two log records.

## 9.5.11 Output of SREPORT 099

Statistics records output by SREPORT 099 are 280-byte fixed-length records with the same layout as records written to the archived system log file. CA-CULPRIT copies the RDW associated with the variable-length records to bytes 1 through 4 of the fixed-length records.

## 9.5.12 SREPORT 000

SREPORT 000 contains the CA-CULPRIT REC parameters that define the fields in each type of statistics record. Each REC parameter defines the start position, length, and data type of a field within the record. The following considerations apply:

- The field names assigned in the CA-CULPRIT report are not always the same as those assigned in the DSECTS.
- Each field defined by a CA-CULPRIT REC parameter has a start position 5 bytes more than in the DSECT.

## 9.6 Producing statistics reports

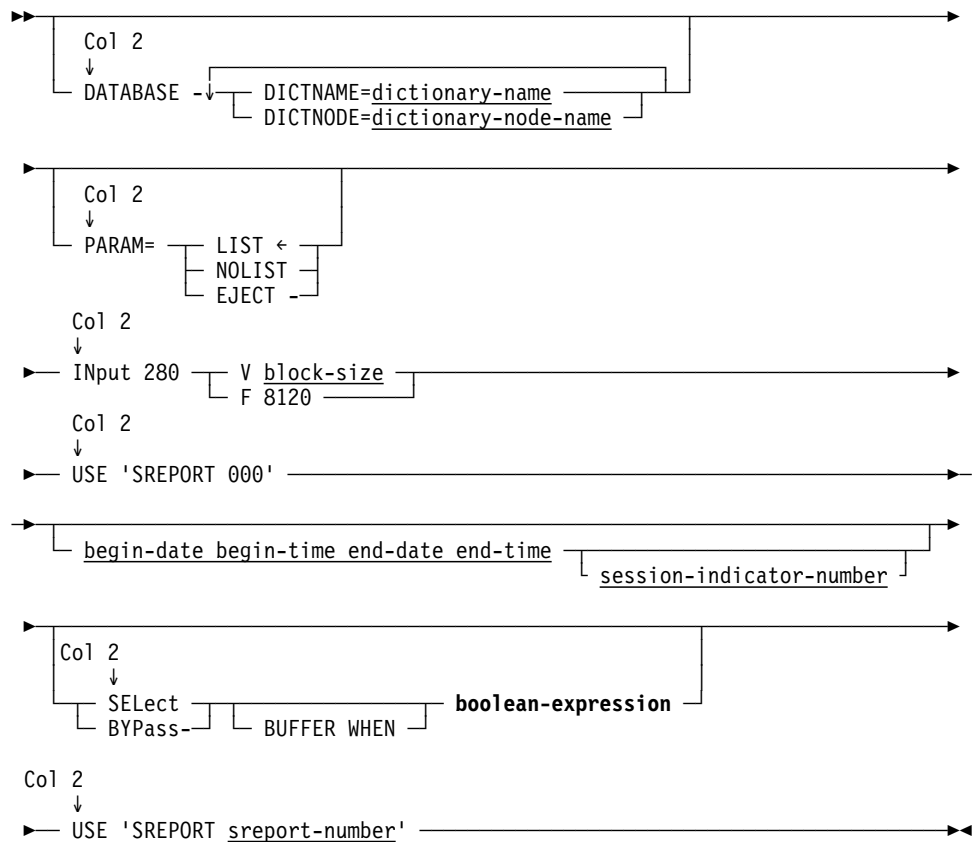
**SYSTEM statements determines how to log statistics:** DC/UCF can log system resource statistics to the database, to a single file, or to alternate files as defined at system configuration by the LOG clause of the SYSTEM statement. When statistics are logged to the database, they are stored in DDLDCLOG, the log area of the data dictionary.

**Reports document statistics logged to database:** Statistics reports document statistics logged to the database. The ARCHIVE LOG utility offloads the statistics from the DDLDCLOG area to an archived system log file; for more information about this utility, see *CA-IDMS Utilities*.

**Input to statistics reports:** Input to the statistics reports is either the archived system log file or a file created by executing SREPORT 099. SREPORT 099 reads the archived system log file and creates a file of archived statistics records.

### 9.6.1 Syntax

Syntax for the CA-CULPRIT parameters is shown below followed by examples of CA-CULPRIT code. Coding is freeform except that each parameter must be coded starting in column 2.



## 9.6.2 Parameters

Syntax rules appear in Chapter 1, "Introduction," except as described below:

### **INput**

Specifies the CA-CULPRIT parameter that designates the physical characteristics of the input file.

### **280**

Specifies the record length, in bytes.

### **V block-size**

Specifies the record type (variable length) and the block size if the archived system log file is used as input:

- In OS/390 systems, the block size is informational
- In VSE/ESA systems, *block-size-n* must match the actual block size of the archived log file (the default block size of the archive file is 6000).

►►For more information about creating output files with CA-IDMS utilities in a VSE/ESA environment, see *CA-IDMS Utilities*.

*Block-size* must be greater than or equal to the actual block size. (The block size for the archived log file is specified in the DMCL.)

### **F 8120**

Specifies the record type (fixed length) and the block size (8120) if the output file produced by SREPORT 099 is used as input.

**Note:** In OS/390 systems, the block size can be omitted.

### **USE 'SREPORT 000'**

Requests SREPORT 000; SREPORT 000 contains CA-CULPRIT REC parameters that define fields used in other statistics report modules. This parameter must be included in all statistics report runs.

### **begin-date**

Specifies the begin date for the period to be covered by the requested reports. If no time period is specified, the requested reports will cover the entire period represented by the input archive file or by the input file from SREPORT 099.

**Note:** A time period and a nonzero session indicator are required for SREPORTs 001, 003, and 012 through 016.

*Begin-date* must be in the Julian form *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day.

### **begin-time**

Specifies the begin time for the period to be covered by the requested reports. *Begin-time* must be in the form *hhmm*, where *hh* is hours based on a 24-hour clock, and *mm* is minutes.

**end-date**

Specifies the end date for the period to be covered by the requested reports.

*End-date* must be in the Julian form *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day.

**end-time**

Specifies the end time for the period to be covered by the requested reports.

*End-time* must be in the form *hhmm*, where *hh* is hours based on a 24-hour clock, and *mm* is minutes.

**session-indicator-number**

Specifies that the requested reports are to cover the indicated occurrence of an DC/UCF session within the specified time period. An DC/UCF session is the period of time from one system startup to the next; the first session within a specified time period begins with the first startup record. SREPORT 000 lists each startup record that exists in the archive file and when the record was logged.

*Session-indicator-number* must be an integer in the range 0 through 9999. If *session-indicator-number* is not specified or is equal to zero, the requested reports will cover the entire period specified.

**SELECT/BYPASS BUFFER WHEN boolean-expression**

Specifies optional selection criteria to be applied during the extract phase of processing. If a SELECT or BYPASS parameter is coded, it must follow the USE 'SREPORT 000' parameter.

**USE 'SREPORT sreport-number'**

Specifies a report module name. *Sreport-number* must be a report number, as specified in the module name. Multiple USE parameters can be included; however, each one must be specified on a separate line.

## 9.6.3 Examples

**Example 1:** SREPORT 003 is requested for the first DC/UCF session occurring between 8:30 and 9:50 PM on March 5, 1999.

```
DATABASE DICTNAME=CULPDICT
INPUT 280 F 8120
USE 'SREPORT 000' (99064 2030 99064 2150 1)
USE 'SREPORT 003'
```

As shown in the output generated for SREPORT 000, the first session within this time frame begins at 20:32 PM with startup record number 5703.

THE FILE BEGINS AT THE FOLLOWING DATE AND TIME: 99064 19:13

RECORD NUMBER	JULIAN DATE	TIME
1,289	99064	19:36
2,415	99064	19:49
5,703	99064	20:32
7,905	99064	21:07

**Example 2:** SREPORT 012 is requested for the second DC/UCF session occurring in the period beginning at 6:00 AM on 10/19/99 and ending at 6:00 PM on 10/19/99. Input consists of the archived system log file created by means of the ARCHIVE LOG utility. Input parameters will not be listed.

```
PARAM=NOLIST
IN 280 V
USE 'SREPORT 000' (99292 0600 99292 1800 2)
USE 'SREPORT 012'
```

**Example 3:** SREPORTs 010 and 005 are requested for all DC/UCF sessions in the period beginning at 8:00 AM on 11/22/99 and ending at 4:00 PM on 11/22/99. Input consists of the output file created by a previous run of SREPORT 099; PS indicates a sequential file. By default, input parameters will be listed.

```
IN 280 F 8120 PS
USE 'SREPORT 000' (99326 0800 99326 1600)
USE 'SREPORT 010'
USE 'SREPORT 005'
```

**Example 4:** Statistics report modules SREPORT 000 and SREPORT 012 are stored in the CULPDICT data dictionary. The SELECT parameter selects only those records stored in the input buffer that specify user ids TAF and TNC. The code for SREPORT 012 identifies TSTUSID as the user-id field name.

```
DATABASE DICTNAME=CULPDICT
IN 280 V
USE 'SREPORT 000'
SELECT BUFFER WHEN TSTUSID EQ ('TAF' 'TNC')
USE 'SREPORT 012'
```

## 9.6.4 Operating system considerations

**JCL coding considerations:** Statistics reports can run either in local mode or under the central version. The JCL to run statistics reports appears in Appendixes A through D for OS/390, VSE/ESA, VM/ESA, and BS2000/OSD operating systems, respectively. The following considerations apply to coding the JCL for running statistics reports:

- The input file containing the statistics records must be defined with ddname/filename/linkname SYS010 (the input file can be either the archived DC/UCF system log file or the output file produced by SREPORT 099).
- When SREPORT 099 is run, the output file must be defined with ddname/filename/linkname SYS020.

**Input file JCL modifications:** For the input file, modify the JCL in Appendixes A through D, as follows:

- **For OS/390 operations systems,** modify ddname SYS010 in Appendix A as follows:

```
//SYS010 DD DSN=user.inputfil,DISP=OLD,UNIT=tape,VOL=SER=nnnnnn
```



---

user.inputfil	data set name of the archived system log file or SREPORT 099 output file
tape	symbolic device name of a disk or tape input file
nnnnnn	volume serial number of the input file

---

- **For VSE/ESA operating systems**, modify filename SYS010 in Appendix B as follows:

```
// TLBL  SYS010,'user.inputfil',,nnnnnn,,f
// ASSGN  SYS010,X'ttt'
```

---

user.inputfil	file-id of the archived system log file or SREPORT 099 output file
nnnnnn	volume serial number of the input file
f	file number of the input file
ttt	device assignment (channel and unit) for tape files (files may be disk instead of tape, in which case a device assignment, DLBL and EXTENT information are also required)

---

- **For VM/ESA operating systems**, modify the SYS010 command in Appendix C, as follows:

- For input from the archived system log file:

```
For tape files: FILEDEF  SYS010 TAP1 SL VOLID nnnnnn
                  (RECFM VB LRECL 280 BLKSIZE bbbb
```

```
For disk files: FILEDEF  SYS010 DISK input file a
                  (RECFM VB LRECL 280 BLKSIZE bbbb
```

---

nnnnnn	volume serial number of the archived system log file
bbbb	block size of the input file
input file a	filename, filetype, filemode of archived system log file

---

- For input from the file created by SREPORT 099:

```
FILEDEF  SYS010 DISK input file a
                  (RECFM FB LRECL 280 BLKSIZE 8120
```

---

input file a	filename, filetype, filemode of archived system log file
--------------	----------------------------------------------------------

---

- **For BS2000/OSD operating systems**, modify linkname SYS010 as follows:

```
/ADD-FILE-LINK L-NAME=SYS010,F-NAME=user.inputfil
```

---

user.inputfil	filename of the archived systems log file or SREPORT 099 output file
---------------	----------------------------------------------------------------------

---

**SREPORT 099 output file JCL modifications:** To create a file of statistics records using SREPORT 099, modify the JCL in Appendices A through D, as follows:

- **For OS/390 operations systems**, modify ddname SYS020 in Appendix A, as follows:

```
//SYS020 DD DSN=user.nonprint,DISP=(NEW,CATLG),
           SPACE=(TRK,(10,10)),UNIT=tape,VOL=SER=nnnnnn
           DCB=(DSORG=PS,RECFM=FB,LRECL=280,BLSIZE=8120)
```

---

user.nonprint	data set name for nonprint output
tape	symbolic device name of the nonprint output file
nnnnnn	volume serial number of the nonprint output file

---

- **For VSE/ESA operating systems** modify file name SYS020 in Appendix B as follows:

```
// ASSGN SYS020,X'ttt'
// TLBL SYS020,'user.nonprint',15
```

---

ttt	device assignment (channel and unit) for tape files (files may be disk instead of tape, in which case a device assignment, DLBL, and EXTENT information are also required.)
user.nonprint,15	file-id and retention period for nonprint/nonpunch output

---

- **For VM/ESA operating systems**, modify the SYS020 command in Appendix C as follows:

```
FILEDEF SYS020 nonprint file a (RECFM FB LRECL 280 BLKSIZE 8120)
```

---

nonprint file a	filename, filetype, filemode of nonprint output
-----------------	-------------------------------------------------

---

- **For BS2000/OSD operating systems**, modify linkname SYS020 in Appendix D as follows:

```
/CREATE-FILE F-NAME=user.nonprint,(SPACE=(primary,secondary)), -
/ SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYS020,F-NAME=user.nonprint
```

---

user.nonprint	filename of nonprint/nonpunch output file
primary	primary space allocation
secondary	secondary space allocation

---

## 9.7 DC/UCF system statistics reports

**Record systemwide data:** System statistics record systemwide data. DC/UCF always collects system statistics (they are not optional) because they require minimal overhead and provide valuable information for tuning and maintaining the DC/UCF system.

**When statistics are logged:** System statistics are logged to the DC/UCF log file at the following times:

- At normal system shutdown
- At the statistics interval established at system generation by the STATISTICS parameter of the SYSTEM statement. The statistics interval can be varied at run time by means of the DCMT VARY STATISTICS command.
- Upon explicit request by means of a DCMT WRITE STATISTICS command.

**Summary of system statistics reports:** System statistics are collected and written for six categories:

General systemwide statistics	SREPORT 003
Systemwide task statistics	SREPORT 012
Systemwide program statistics	SREPORT 013
Systemwide queue statistics	SREPORT 014
Systemwide line statistics	SREPORT 015
Systemwide physical terminal (including UCF PTERMs) statistics	SREPORT 016

### 9.7.1 SREPORT 003 — IDMS-DC System Statistics

**Contents:** SREPORT 003, the IDMS-DC System Statistics report, summarizes all systemwide statistics and thus provides an overview of system performance.

REPORT NO. 03	IDMS-DC SYSTEM STATISTICS R15.0	09/08/99 PAGE 1
SELECTED FROM: 00000 00:00	TO: 00000 00:00	
ACTUAL: 99249 03:10	TO: 99249 04:39	
SYSTEM STATISTICS		
124 TOTAL TASKS	0 STD PGMPPOOL LOADS	
69 TOTAL SYSTEM TASKS	0 STD PGMPPOOL WAITS	
4 TASKS ABENDED	0 STD PGM PAGES LOADED	
0 RUNAWAY TASKS ABORTED	26 RENTPOOL LOADS	
0 TIMES AT MAX TASK	0 RENTPOOL WAITS	
0 SHORT ON STORAGE	724 RENT PGM PGS LOADED	
0 OVER RLE THRESH	1 XA PGMPPOOL LOADS	
0 OVER RCE THRESH	0 XA PGMPPOOL WAITS	
0 OVER DPE THRESH	5 XA PGM PGS LOADED	
0 OVER ILE THRESH	157 XA RENTPOOL LOADS	
0 STORAGE POOL WAITS	0 XA RENTPOOL WAITS	
1,317 STG REQS - PASS 1	4,247 XA RENT PGS LOADED	
495 STG REQS - PASS 2	690 PAGE RELEASE RQSTS	
0 PUT JOURNALS	1,061 PAGES RELEASED	
0 SET TIME WAITS	0 PAGE FIX RQSTS	
297 SET TIME POSTS	0 PAGES PFIXED	
0 SET TIME STRTTASKS	0 PAGE FREE RQSTS	
294 SET TIME CANCELS	0 PAGES PGFREED	
0 AUTOSTART TASKS		
DC STATISTICS		
.5001 USER MODE CPU TIME	2.0043 SYSTEM MODE CPU TIME	
2,961 DC SERVICE REQUESTS	147 GET SCRATCHES	
1,097 DB SERVICE REQUESTS	196 PUT SCRATCHES	
532 PROGRAMS CALLED	149 DELETE SCRATCHES	
367 MAX # RLE'S USED	22 GET QUEUES	
314 MAX # RCE'S USED	7 PUT QUEUES	
321 MAX # DPE'S USED	7 DELETE QUEUES	
594 STACK HI WATERMARK	1,658 GET TIMES	
1,812 GET STORAGES	592 SET TIMES	
1,572 FREE STORAGES		
DB STATISTICS		
1,561 PAGES REQUESTED	5 CALC RECS NO OFLOW	
1,169 PAGES READ	0 CALC RECS OFLOW	
13 PAGES WRITTEN	7 VIA RECS NO OFLOW	
1,226 CALLS TO DBMS	0 VIA RECS OFLOW	
807 RECORDS REQUESTED	0 FRAGMENTS STORED	
451 RECORDS CURRENT OF RU	0 RECORDS RELOCATED	
149 TOTAL LOCKS	0 RECORDS UPDATED	
0 PAGES FOUND IN CACHE	0 PAGES IN PREFETCH BUFF	
INDEX STATISTICS		
0 SR8 SPLITS	0 SR8 STORES	
0 SR8 SPAWNS	0 SR8 ERASES	
0 ORPHANS ADOPTED	0 SR7 STORES	
0 BTREE SEARCHES	0 SR7 ERASES	
0 MIN LEVELS SEARCHED	0 TOTAL LEVELS SEARCHED	
0 MAX LEVELS SEARCHED		
SQL STATISTICS		
0 SQL COMMANDS	0 TUPLES FETCHED	
0 SORTS	0 ROWS INSERTED	
0 TUPLES SORTED	0 ROWS UPDATED	
0 MIN SORT	0 ROWS DELETED	
0 MAX SORT	0 AM RECOMPILES	

Figure 9-1. Sample SREPORT 003

**Field descriptions:** A description of the fields in the IDMS-DC System Statistics report follows:

#### SELECTED FROM/TO

Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

**ACTUAL/TO**

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

**TOTAL TASKS**

Indicates the total number of system and user tasks, including external request units, executed for this session.

**TOTAL SYSTEM TASKS**

Indicates the number of system tasks active at the conclusion of this session. System-initiated tasks include FACTOTUM, MASTER, DBRC, print, and line drivers.

**TASKS ABENDED**

Indicates the number of tasks that abended during this session.

**RUNAWAY TASKS ABORTED**

Indicates the number of tasks that terminated abnormally because their execution time exceeded the maximum amount of time defined in the RUNAWAY INTERVAL clause of the SYSTEM statement.

**TIMES AT MAX TASK**

Indicates the number of times a new task could not start because the system was processing the maximum number of tasks specified at system generation in the MAXIMUM ERUS and MAXIMUM TASK clauses of the SYSTEM statement.

Interpretation: Generally, this value should be close to 0. Values equal to 0 may indicate that the overall system size is larger than necessary; values greater than 0 may represent peaks in system loads or chronic system overload.

To reduce the number of times at maximum tasks, make one or more of the following system adjustments:

- Increase the MAX ERUS and MAX TASKS thresholds
- Decrease the limit specified for the MAXIMUM CONCURRENT THREADS clause of the TASK statement at system generation

**SHORT ON STORAGE**

Indicates the number of times program storage requests were not satisfied. Generally, the value should be close to zero; values greater than zero should represent peaks in storage utilization, rather than chronic shortages.

Interpretation: Values consistently greater than zero indicate the size of the storage pool or storage cushion is too small. The size of the storage pool is defined at system generation with the STORAGE POOL and XA STORAGE POOL clauses of the SYSTEM system generation statement; the size of the cushion is defined with the CUSHION clause of the same statement.

**OVER RLE THRESH**

Indicates how many times the number of resource link elements (RLEs) exceeded the value allocated in the SYSTEM statement at system generation. When the threshold is reached, the task that is executing abends; if the task is a system task, the system abends. Ideally, this value should be 0.

#### **OVER RCE THRESH**

Indicates how many times the number of resource control elements (RCEs) exceeded the value allocated in the SYSTEM statement at system generation. When the threshold is reached, the task that is executing abends; if the task is a system task, the system abends. Ideally, this value should be 0.

#### **OVER DPE THRESH**

Indicates how many times the number of deadlock prevention elements (DPEs) exceeded the value allocated in the SYSTEM statement at system generation. When the threshold is reached, the task that is executing abends; if the task is a system task, the system abends. Ideally, this value should be 0.

#### **OVER ILE THRESH**

Indicates the number of time the internal lock elements (ILEs) exceeded their allocated value. When the threshold is reached, the task that is executing abends. If the task is a system task, the system abends. Ideally, this value should be zero (0).

#### **STORAGE POOL WAITS**

Indicates the number of times tasks had to wait for a sufficient amount of contiguous storage. This value should be as low as possible; if the value is large or increasing, increase the size of the storage cushion in the CUSHION clause of the STORAGE POOL statement at system generation.

#### **STG REQS - PASS 1**

Indicates the number of storage requests that allocated space on a previously allocated page.

#### **STG REQS - PASS 2**

Indicates the number of storage requests that allocated space on a previously allocated page and on a contiguous new page.

#### **SET TIME WAITS**

Indicates the number of program requests to place a task in a wait state. The program issues the request with a SET TIMER WAIT DML statement.

#### **SET TIME POSTS**

Indicates the number of program requests to post a user-specified event control block (ECB) after the specified time interval elapses. The program issues the request with a SET TIMER POST DML statement.

#### **SET TIME STRTTASKS**

Indicates the number of program requests to initiate a user-specified task after the specified time interval elapses. The program issues the request with a SET TIMER START DML statement.

#### **SET TIME CANCELS**

Indicates the number of program requests to cancel the effect of a previously issued SET TIMER request. The program issues the request with a SET TIMER CANCEL DML statement.

#### **AUTOSTART TASKS**

(DC/UCF only) Indicates the number of times the associated task of each queue was invoked to process queue records. The task is invoked each time the queue

threshold is exceeded. The threshold is defined with the THRESHOLD clause of the QUEUE system generation statement; the task is identified by the INVOKES TASK clause of the same statement.

**STD PGMPOOL LOADS**

Indicates the number of nonresident modules loaded into the 24-bit program pool; nonresident modules include programs, CA-ADS and ADS/Batch dialogs, maps (DC/UCF only), and tables.

**STD PGMPOOL WAITS**

Indicates the number of times program loads were delayed due to insufficient space in the 24-bit program pool. Ideally, this value should be zero; a value greater than zero indicates insufficient space in the program pool for the volume of program load activity in the system.

Interpretation: If this value is high, the size of the 24-bit program pool should be increased or nonresident programs should be redefined as either resident or reentrant. Program pool usage can be observed dynamically by means of the OPER WATCH PR command. For more information about program pools, see *CA-IDMS System Operations*.

**STD PGM PAGES LOADED**

Indicates the number of pages used by programs loaded into the 24-bit program pool. The size of a page equals 4K. Only one program can occupy a program pool page; that is, a 4.5K program uses two pages of program pool.

**RENTPOOL LOADS**

Indicates the number of reentrant programs loaded into the 24-bit reentrant pool; reentrant programs include CA-ADS dialogs and subschemas.

**RENTPOOL WAITS**

Indicates the number of times reentrant program loads were delayed due to insufficient space in the 24-bit reentrant pool. Ideally, this value should be zero; a value greater than zero indicates the size of the reentrant pool should be increased.

**RENT PGM PGS LOADED**

Indicates the number of pages used by reentrant programs loaded into the 24-bit reentrant pool. A page is 512 bytes (0.5K).

**XA PGMPOOL LOADS**

(Extended addressing only) Indicates the number of nonresident modules loaded into the 31-bit XA program pool; nonresident modules include programs, subschemas, maps, database procedures, and tables that have been assigned an RMODE of ANY.

**XA PGMPOOL WAITS**

(Extended addressing only) Indicates the number of times program loads were delayed due to insufficient space in the 31-bit XA program pool. Ideally, this value should be zero; a value greater than zero indicates insufficient space in the XA program pool for the volume of program load activity in the system.

**XA PGM PGS LOADED**

(Extended addressing only) Indicates the number of pages used by nonresident modules loaded into the 31-bit XA program pool. The size of a page equals 4K.

Only one program can occupy a program pool page; that is, a 4.5K program uses two pages of program pool.

**XA RENTPOOL LOADS**

(Extended addressing only) Indicates the number of reentrant modules loaded into the 31-bit XA reentrant pool; reentrant modules include reentrant programs, subschemas, CA-ADS dialogs, and DC/UCF maps that have been assigned an RMODE of ANY.

**XA RENTPOOL WAITS**

(Extended addressing only) Indicates the number of times program loads were delayed due to insufficient space in the 31-bit XA reentrant pool. Ideally, this value should be zero; a value greater than zero indicates insufficient space in the XA reentrant pool for the volume of program load activity in the system.

**XA RENT PGS LOADED**

(Extended addressing only) Indicates the number of pages used by reentrant modules loaded into the 31-bit XA reentrant pool. The size of a page equals 512 bytes (0.5K).

**PAGE RELEASE RQSTS**

Indicates the number of requests to release 4K virtual pages.

**PAGES RELEASED**

Indicates the number of 4K byte pages actually released.

**PAGE FIX RQSTS**

(VS systems only) Indicates the number of requests to fix 4K virtual pages allocated to storage pools defined to the system.

**PAGES PFIXED**

(VS systems only) Indicates the number of 4K virtual pages fixed in storage pools defined to the system.

**PAGE FREE RQSTS**

Indicates the number of requests to release 4K fixed virtual pages allocated to storage pools defined to the system, thereby making storage eligible for paging out.

**PAGES PGFREED**

Indicates the number released of 4K fixed virtual pages allocated to system storage pools.

**PUT JOURNALS**

Indicates the number of program requests to write statistics to the journal file with the WRITE JOURNAL DML statement.

**USER MODE CPU TIME**

Indicates the amount of CPU time (in ten-thousandths seconds) spent in executing user code.

**DC SERVICE REQUESTS**

Indicates the number of times user programs requested DC/UCF services (for example, GET STORAGE requests). This value also includes both explicit and implicit requests for database services. For example, OBTAIN *record-name* is an



explicit program request. PUT QUEUE is an implicit request for database services because the DC/UCF system must store the queue record.

**DB SERVICE REQUESTS**

Indicates the number of times a user or system program requests database services (for example, OBTAIN *record-name*).

For LRF and SQL programs, DB SERVICE REQUESTS should be less than or equal to the number of CALLS TO DBMS. You can use these values to evaluate how efficiently the LRF or SQL path extracts data. For example, a program OBTAIN *logical-record* command increments DB SERVICE REQUESTS by one, but may greatly increment the CALLS TO DBMS value, especially if an area sweep occurs due to the NULL SELECT clause.

**PROGRAMS CALLED**

Indicates the number of programs called (for example, the number of #LOAD PGM requests issued).

**MAX # RLE'S USED**

Indicates the highest number of resource link elements (RLEs) used during this session. If this value approximates the threshold established at system generation, increase the threshold.

**MAX # RCE'S USED**

Indicates the highest number of resource control elements (RCEs) used during this session. If this value approximates the threshold established at system generation, increase the threshold.

**MAX # DPE'S USED**

Indicates the highest number of deadlock prevention elements (DPEs) used during this session. If this value approximates the threshold established at system generation, increase the threshold.

**STACK HI WATERMARK**

Indicates the largest amount of the task control element (TCE) stack area used by any task. The stack size is defined at system generation with the STACKSIZE clause of the SYSTEM statement.

**GET STORAGES**

Indicates the number of program requests to acquire variable storage dynamically from an DC/UCF storage pool or obtain the address of a previously acquired storage area. The program issues the request with the GET STORAGE DML statement.

**FREE STORAGES**

Indicates the number of program requests to free all or part of an DC/UCF storage area. The program issues the request with the FREE STORAGE DML statement.

**SYSTEM MODE CPU TIME**

Indicates the amount of CPU time (in ten-thousandths seconds) spent in executing system code.

#### **GET SCRATCHES**

Indicates the number of program requests for scratch records from the DDLDCSCR area. The program issues the request with the GET SCRATCH DML statement.

#### **PUT SCRATCHES**

Indicates the number of program requests to store or replace a scratch record in the DDLDCSCR area. The program issues the request with the PUT SCRATCH DML statement.

#### **DELETE SCRATCHES**

Indicates the number of program requests to delete scratch records from the DDLDCSCR area. The program issues the request with the DELETE SCRATCH DML statement.

#### **GET QUEUES**

Indicates the number of program requests to retrieve a queue record from the DDLDCRUN area and place it in a storage area associated with the issuing program. The program issues the request with the GET QUEUE DML statement.

#### **PUT QUEUES**

Indicates the number of program requests to store a queue record in the DDLDCRUN area. The program issues the request with the PUT QUEUE DML statement.

#### **DELETE QUEUES**

Indicates the number of program requests to delete queue records from the DDLDCRUN area. The program issues the request with the DELETE QUEUE DML statement.

#### **GET TIMES**

Indicates the number of program requests for the system date and time. The program issues the request with the GET TIME DML statement.

#### **SET TIMES**

Indicates the number of SETTIME requests to define an event that is to occur after a specified time interval.

#### **PAGES REQUESTED**

Indicates the number of pages requested by IDMSDBMS (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool.

Interpretation: The ratio of PAGES REQUESTED/PAGES READ is the **buffer utilization ratio**. It measures the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio the better. Ratios consistently below 2.0 indicate that processing is random or that the buffer-pool size is too small.

The buffer utilization ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

**PAGES READ**

Indicates the number of database pages read from disk.

**PAGES WRITTEN**

Indicates the number of database pages physically written to disk. A page can be updated several times before it is actually written back to the database.

**CALLS TO DBMS**

Indicates the number of calls to the database management system.

**Note:** Execution of each navigational DML request involves one call; execution of each logical record facility (LRF) and SQL request typically involves multiple calls.

**RECORDS REQUESTED**

Indicates the number of database records requested by IDMSDBMS.

Interpretation: The ratio of RECORDS REQUESTED to PAGES READ is the **space management ratio**. The space management ratio measures how well space is allocated (for example, VIA options, CALC distribution, and buffering). The higher the ratio the better. Ratios less than 4 or less than the norm indicate that the size of the buffer should be increased and database tuning should be performed.

The space management ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

**RECORDS CURRENT OF RU**

Indicates the number of records made current of transaction.

Interpretation: The ratio of RECORDS REQUESTED to RECORDS CURRENT is the **effectiveness ratio**. The effectiveness ratio measures the amount of work CA-IDMS/DB is doing for the programmer (that is, how many records the DBMS has to examine to find the one requested). The lower the ratio the better. If the ratio is high, examine set options (for example, sorted order or next pointers only) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

**TOTAL LOCKS**

Indicates the number of all locks acquired and released by all completed transactions. This is NOT a count of locks currently held.

**PAGES FOUND IN CACHE**

Indicates the number of requested database pages that have been found in a shared cache or an ESA dataspace.

**CALCRECS NO OFLOW**

Indicates the number of CALC records stored on the target page.

**CALC RECS OFLOW**

Indicates the number of CALC records not stored on the target page.

Interpretation: The ratio of CALC records stored on their target page to the total

number stored (that is, hits plus overflows) is the **CALC cluster ratio**. The ratio reflects the efficiency of the CALC algorithm.

The CALC cluster ratio is especially important when the database is loaded or restructured. Ideally, the ratio should be 1, which indicates no overflow. Ratios consistently less than 1 or less than the norm indicate that space utilization is getting high and the database should be tuned.

**VIA RECS NO OFLOW**

Indicates the number of VIA and/or DIRECT records stored on the target page.

**VIA RECS OFLOW**

Indicates the number of VIA and/or DIRECT records not stored on the target page.

Interpretation: The ratio of VIA records stored on their target page to the total number of VIA records stored (that is, hits plus overflows) is the **VIA cluster ratio**. The ratio reflects how well VIA records cluster around their owner.

Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate very large data clusters, high utilization of space, or small page size.

**FRAGMENTS STORED**

Indicates the number of noncontiguous segments (fragments) stored for variable-length records.

**RECORDS RELOCATED**

Indicates the number of records relocated from their home page.

**PAGES IN PREFETCH BUFF**

Indicates the number of database pages that have been directly found in a prefetch buffer.

**SR8 SPLITS**

Indicates the number of SR8 splits.

**SR8 SPAWNS**

Indicates the number of SR8 spawns.

**ORPHANS ADOPTED**

Indicates count of Index members or SR8s whose up-level pointers were corrected to point to the actual SR8 in which they appear.

**BTREE SEARCHES**

Indicates number of Btree index probes.

**MIN LEVELS SEARCHED**

Indicates least number of levels descended.

**MAX LEVELS SEARCHED**

Indicates highest number of levels descended.

**SR8 STORES**

Indicates count of SR8s created.

**SR8 ERASES**

Indicates count of SR8s erased.

**SR7 STORES**

Indicates count of SR7s created.

**SR7 ERASES**

Indicates count of SR7s erased.

**TOTAL LEVELS SEARCHED**

Indicates total number of levels descended.

**SQL COMMANDS**

Indicates the number of SQL commands executed.

**SORTS**

Indicates the number of SQL sorts performed.

**TUPLES SORTED**

Indicates the number of rows participating in all sorts.

**MIN SORT**

Indicates the least number of rows sorted.

**MAX SORT**

Indicates the largest number of rows sorted.

**TUPLES FETCHED**

Indicates number of tuples FETCHed.

**ROWS INSERTED**

Indicates the number of rows INSERTed.

**ROWS UPDATED**

Indicates the number of rows UPDATed.

**ROWS DELETED**

Indicates the number of rows DELETED.

**AM RECOMPILES**

Indicates the number of automatic access module recompilations.

For a detailed explanation of DML commands, see *CA-IDMS Navigational DML Programming*. For more information about system generation statements, see *CA-IDMS System Generation*.

## 9.7.2 SREPORT 012 — IDMS-DC Task Summary

**Contents:** SREPORT 012, the IDMS-DC Task Summary report, summarizes systemwide task statistics, indicating the number of times each task was invoked. Systems administrators can use SREPORT 012 to monitor trends in task usage; for example, watching for peaks in task usage over a set time period.

The figure below shows one page of a sample IDMS-DC Task Summary report:

REPORT NO. 12	IDMS-DC TASK SUMMARY R15.0				09/08/99 PAGE 3
SELECTED FROM: 00000 00:00	TO: 00000 00:00				
ACTUAL: 99249 17:03	TO: 99250 08:18				
TASK	TIMES INVOKED				
S	0				
SCHEMA	0				
SCHEMAT	0				
SEND	0				
SHOWMAP	0				
SIGNOFF	0				
SIGNON	0				
SSC	0				
SSCT	0				
SUSPEND	0				
SYSGEN	0				
SYSGENT	0				
TCF	0				
TOTAL	4				

Figure 9-2. Sample SREPORT 012

**Field descriptions:** A description of the fields in the IDMS-DC Task Summary report follows:

#### SELECTED FROM/TO

Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

#### ACTUAL/TO

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

#### TASK

Specifies the task identifier that is used at run time by a terminal operator or program to invoke the task. A task is the basic unit of work under DC/UCF. It consists of a main program and one or more additional programs. A task is identified to the system by a unique name (such as OLM) that is usually identical to the task code used by the teleprocessing system.

#### TIMES INVOKED

Specifies the number of times the task was invoked for the session.

### 9.7.3 SREPORT 013 — IDMS-DC Program Summary

**Contents:** SREPORT 013, the IDMS-DC Program Summary report, summarizes systemwide program statistics. System administrators can use SREPORT 013 to monitor program activity relative to available storage.

**Available program pools:** A program can be assigned to any one of the following pools:

- 24-bit program pool
- 24-bit reentrant pool
- 31-bit program pool (Extended addressing only)
- 31-bit reentrant pool (Extended addressing only)

**Note:** A program can be put into both a 24-bit pool and a 31-bit pool depending on the LOC= parameter used to invoke the task.

The figure below shows one page of a sample IDMS-DC Program Summary report:

REPORT NO. 13			IDMS-DC PROGRAM SUMMARY R15.0			09/08/99 PAGE 40		
SELECTED FROM: 00000 00:00			TO: 00000 00:00					
ACTUAL: 99249 17:03			TO: 99250 08:18					
PROGRAM NAME	PGM VER	DICTNAME	DICTNODE	TIMES CALLED	TIMES LOADED	RATIO OF CALLED/LOADED	TIMES WAITED	TIMES CHECKED
RHDCWAIT	1			0	1	0.0000		
RHDCWTL	1			0	1	0.0000		
RM000121	1	ASFDICT		1	1	1.0000		
RM000122	1	ASFDICT		1	1	1.0000		
RM000123	1	ASFDICT		1	1	1.0000		
RM000124	1	ASFDICT		18	4	4.5000		
RU000121	1	ASFDICT		3	3	1.0000		
RU000122	1	ASFDICT		3	3	1.0000		
RU000123	1	ASFDICT		3	3	1.0000		
RU000124	1	ASFDICT		20	8	2.5000		
TSTDNKA	1			42	1	42.0000		
TSTDNWKS	1			2	1	2.0000		
UMBRMAP	1			0	0	0.0000		
UMBRMAPE	1			0	0	0.0000		
XASFNWKS	1			1	1	1.0000		
TOTALS				1,458	181	8.0552	0	0

Figure 9-3. Sample SREPORT 013

**Field descriptions:** A description of the fields in the IDMS-DC Program Summary report follows:

**SELECTED FROM/TO**

Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

**ACTUAL/TO**

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

**PROGRAM NAME**

Specifies the name of the program load module.

**PGM VER**

Indicates the version number associated with the named program.

**DICTNAME**

Names the dictionary in which the program resides.

**DICTNODE**

Names the DC/UCF system that controls the dictionary in which the program resides.

**TIMES CALLED**

Indicates the number of times the program was called.

**TIMES LOADED**

Indicates the number of times the program was loaded from disk to a pool.

Interpretation: The ratio of TIMES CALLED to TIMES LOADED measures the effectiveness of the program pool size. The higher the ratio the better. A low ratio for a frequently called program indicates that the size of the program pool should be enlarged or that the program should be made resident, reentrant, or reusable.

**RATIO OF CALLED/LOADED**

Measures the effectiveness of the program pool size. The higher the ratio the better. A low ratio for a frequently called program indicates that the size of the program pool should be enlarged or that the program should be made resident, reentrant, or reusable.

**TIMES WAITED**

Indicates the number of times the program waited to be loaded.

**TIMES CHECKED**

Indicates the number of program check errors that occurred.



## 9.7.4 SREPORT 014 — IDMS-DC Queue Summary

**Contents:** SREPORT 014, the IDMS-DC Queue Summary report, summarizes systemwide queue statistics. System administrators can use SREPORT 014 to monitor queue activity.

REPORT NO. 14	IDMS-DC QUEUE SUMMARY R15.0	09/08/99 PAGE 1
SELECTED FROM: 00000 00:00	TO: 00000 00:00	
ACTUAL: 99249 17:03	TO: 99250 08:18	
QUEUE NAME	TIMES AUTOTASK STARTED	
OLQQNOTE	0	
TOTAL	0	

Figure 9-4. Sample SREPORT 014

**Field descriptions:** A description of the fields in the IDMS-DC Queue Summary report follows:

### SELECTED FROM/TO

Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

### ACTUAL/TO

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

### QUEUE NAME

Specifies the queue identifier. A queue is a database work area shared by tasks on all DC/UCF terminals and by batch programs. Queue records can be transferred between tasks or applications or from one terminal to another.

### TIMES AUTOTASK STARTED

Indicates the number of times each queue's associated task was invoked to process queue records. If the value is consistently zero, review the threshold level assigned in the THRESHOLD IS clause of the QUEUE system generation statement; for more information, see *CA-IDMS System Generation*.

## 9.7.5 SREPORT 015 — IDMS-DC Line Summary

**Contents:** SREPORT 015, the IDMS-DC Line Summary report, summarizes systemwide line statistics. Systems administrators can use SREPORT 015 to monitor the quality of lines in the system.

REPORT NO. 15		IDMS-DC LINE SUMMARY R15.0		09/08/99 PAGE 1	
SELECTED FROM: 00000 00:00		TO: 00000 00:00			
ACTUAL: 99249 17:03		TO: 99250 08:18			
LINE NAME	READS	WRITES	READ ERRORS	WRITE ERRORS	
CONSOLE	0	0			
CTCQA10	0	0			
CTCQA21	0	0			
CTCQA35	0	0			
JESRDR	0	0			
S3270Q1	0	0			
UCFLINE	0	0			
VTAM91	0	0			
TOTALS	0	0	0	0	

Figure 9-5. Sample SREPORT 015

**Field descriptions:** A description of the fields in the IDMS-DC Line Summary report follows:

**SELECTED FROM/TO**

Specifies the time period specified on the USE 'SREPORT 000' parameter where **SELECTED FROM** specifies the beginning date and time and **SELECTED TO** specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

**ACTUAL/TO**

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, **ACTUAL** and **TO** represent the time range for the entire input archive file or input file created by SREPORT 099.

**LINE NAME**

Specifies the line identifier. A line is a system component that defines the method of communication for physical terminals that use the same access method.

**READS**

Indicates the number of reads performed for each line.

**WRITES**

Indicates the number of writes performed for each line.

**READ ERRORS**

Indicates the number of read errors that occurred for each line. A large value indicates problems in the line.

**WRITE ERRORS**

Indicates the number of write errors that occurred for each line. A large value indicates problems in the line.

## 9.7.6 SREPORT 016 — IDMS-DC Physical Terminal Summary

**Contents:** SREPORT 016, the IDMS-DC Physical Terminal Summary report, summarizes systemwide line statistics.

REPORT NO. 16		IDMS-DC PHYSICAL TERMINAL SUMMARY R15.0				09/08/99 PAGE		2
SELECTED FROM: 00000 00:00		TO: 00000 00:00						
ACTUAL: 99249 17:03		TO: 99250 08:18						
PHYSICAL TERMINAL	READS	WRITES	READ ERRORS	WRITE ERRORS	RESPONSES	CUMULATIVE RESPONSE TIME	AVERAGE RESPONSE TIME	CUMULATIVE I/O TIME
VP91007	0	0			0	0.0000	0.0000	0.0000
VP91008	0	0			0	0.0000	0.0000	0.0000
VP91009	0	0			0	0.0000	0.0000	0.0000
VP91010	0	0			0	0.0000	0.0000	0.0000
VP91011	0	0			0	0.0000	0.0000	0.0000
VP91012	0	0			0	0.0000	0.0000	0.0000
VP91013	0	0			0	0.0000	0.0000	0.0000
VP91014	0	0			0	0.0000	0.0000	0.0000
VP91015	0	0			0	0.0000	0.0000	0.0000
TOTALS	312	238	0	0	0			

Figure 9-6. Sample SREPORT 016

**Field descriptions:** A description of the fields in the IDMS-DC Physical Terminal Summary report follows:

### SELECTED FROM/TO

Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

### ACTUAL/TO

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

### PHYSICAL TERMINAL

Specifies the physical terminal identifier. A physical terminal is a physical device such as a CRT (3270-type device), TTY, or printer that exists within a teleprocessing system.

### READS

Indicates the number of reads performed for each physical terminal.

### WRITES

Indicates the number of writes performed for each physical terminal.

**READ ERRORS**

Indicates the number of read errors that occurred for each physical terminal.

**WRITE ERRORS**

Indicates the number of write errors that occurred for each physical terminal.

**RESPONSES**

Indicates the number of physical terminal responses. A response begins when a physical terminal issues a read request and ends when the next read request is issued.

**CUMULATIVE RESPONSE TIME**

Indicates the cumulative response time, in hundredths of a second, for the physical terminal. Response time is the total non-I/O time measured from the one read request to the next.

**AVERAGE RESPONSE TIME**

Specifies the ratio of CUMULATIVE RESPONSE TIME to RESPONSES. Systems administrators can monitor this value to determine the impact of system configuration changes upon response time.

**CUMULATIVE I/O TIME**

Specifies the cumulative I/O time, in hundredths of a second, for the physical terminal.

## 9.8 Task and External Request Unit Service (ERUS) statistics reports

**Overview:** Task statistics and external request unit service (ERUS) statistics record the resource usage of individual tasks and external request units. The following considerations apply:

- Task and ERUS statistics, including database statistics, incorporate statistics of system run units (RHDCRUAL/IDMSXTAL) when these run units do work for the task. For example, loading a load module from a load area requires database requests which show up in the task statistics.
- Task and ERUS statistics include system-mode statistics; that is, when DC/UCF does work for the task. For example, when a program issues a BIND RUN UNIT, the system obtains variable storage; the storage requests show up in the task statistics.

**Uses for task and ERUS statistics:** Task and ERUS statistics are useful for monitoring and tuning individual application programs.

**Statistics collected only upon request:** DC/UCF collects task and ERUS statistics only when requested to do so by the user, because the statistics require additional overhead and, when written to the log file, generate a large volume of data.

►► For more information about collecting these statistics, see *CA-IDMS System Operations*.

**Summary of task statistics reports:** Five statistics reports summarize task and external request unit activity:

---

SREPORT 005	Summarizes task statistics by user identifier.
SREPORT 006	Summarizes task statistics by logical terminal identifier.
SREPORT 007	Summarizes task statistics by task code.
SREPORT 008	Summarizes ERUS statistics by accounting data.
SREPORT 009	Summarizes ERUS statistics by program name.

---

## 9.8 Task and External Request Unit Service (ERUS) statistics reports

REPORT NO. 05		IDMS-DC TASK STATISTICS BY USER ID R15.0		09/08/99 PAGE 2	
SELECTED FROM: 98001 00:00		TO: 98365 23:59			
ACTUAL: 98300 15:11		TO: 98300 15:25			
USER ID: JEJ					
14	NUMBER TASK EXECUTIONS				
86	NUMBER PROGRAMS CALLED	19	NUMBER PROGRAMS LOADED		
3	NUMBER TERMINAL READS	9	NUMBER TERMINAL WRITES		
0	NUMBER TERMINAL ERRORS	1,115	NUMBER GETSTG REQUESTS		
774	NUMBER GETSCR REQUESTS	787	NUMBER PUTSCR REQUESTS		
766	NUMBER DELSCR REQUESTS	0	NUMBER GETQUE REQUESTS		
0	NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS		
222	NUMBER GETTIME REQUESTS	0	NUMBER SETTIME REQUESTS		
1,689	NUMBER DB SERVICES RQSTS	949	NUMBER PAGES READ		
.5859	TASK USER MODE TIME	.7812	TASK SYSTEM MODE TIME		
26.6278	TASK WAIT TIME				
20	NUMBER PAGES WRITTEN	3,720	NUMBER PAGES REQUESTED		
2	NUMBER CALC RECS NO OFLOW	0	NUMBER CALC RECS OFLOW		
120	NUMBER VIA RECS NO OFLOW	0	NUMBER VIA RECS OFLOW		
5,562	NUMBER RECORDS REQUESTED	1,782	NUMBER RECS CURRENT OF RU		
0	NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED		
2,649	NUMBER CALLS TO DBMS	7,466	TOTAL LOCKS ACQUIRED		
56	NUMBER SR8 SPLITS	44	NUMBER SR8 STORES		
4	NUMBER SR8 SPAWNS	44	NUMBER SR8 ERASES		
18	NUMBER ORPHAN ADOPT	2	NUMBER SR7 STORES		
381	NUMBER BTREE SEARCHES	2	NUMBER SR7 ERASES		
27	NUMBER SQL COMMANDS	1	NUMBER SQL SORTS		
785	NUMBER TUPLES FETCHED	728	NUMBER TUPLES SORTED		
0	NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES		
0	NUMBER ROWS UPDATED				
0	NUMBER ROWS DELETED				

Figure 9-7. Sample SREPORT 005

REPORT NO. 06		IDMS-DC TASK STATISTICS BY LTERM ID R15.0		09/08/99 PAGE 3	
SELECTED FROM: 98001 00:00		TO: 98365 23:59			
ACTUAL: 98300 15:11		END: 98300 15:25			
LTERM ID: VL71001					
14	NUMBER TASK EXECUTIONS				
86	NUMBER PROGRAMS CALLED	19	NUMBER PROGRAMS LOADED		
3	NUMBER TERMINAL READS	9	NUMBER TERMINAL WRITES		
0	NUMBER TERMINAL ERRORS	1,115	NUMBER GETSTG REQUESTS		
774	NUMBER GETSCR REQUESTS	787	NUMBER PUTSCR REQUESTS		
766	NUMBER DELSCR REQUESTS	0	NUMBER GETQUE REQUESTS		
0	NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS		
222	NUMBER GETTIME REQUESTS	0	NUMBER SETTIME REQUESTS		
1,689	NUMBER DB SERVICE RQSTS	949	NUMBER PAGES READ		
.5859	TASK USER MODE TIME	.7812	TASK SYSTEM MODE TIME		
26.6278	TASK WAIT TIME				
20	NUMBER PAGES WRITTEN	3,720	NUMBER PAGES REQUESTED		
2	NUMBER CALC RECS NO OFLOW	0	NUMBER CALC RECS OFLOW		
120	NUMBER VIA RECS NO OFLOW	0	NUMBER VIA RECS OFLOW		
5,562	NUMBER RECORDS REQUESTED	1,782	NUMBER RECS CURRENT OF RU		
0	NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED		
2,649	NUMBER CALLS TO DBMS	7,466	TOTAL LOCKS ACQUIRED		
56	NUMBER SR8 SPLITS	44	NUMBER SR8 STORES		
4	NUMBER SR8 SPAWNS	44	NUMBER SR8 ERASES		
18	NUMBER ORPHAN ADOPT	2	NUMBER SR7 STORES		
381	NUMBER BTREE SEARCHES	2	NUMBER SR7 ERASES		
27	NUMBER SQL COMMANDS	1	NUMBER SQL SORTS		
785	NUMBER TUPLES FETCHED	728	NUMBER TUPLES SORTED		
0	NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES		
0	NUMBER ROWS UPDATED				
0	NUMBER ROWS DELETED				

Figure 9-8. Sample SREPORT 006

REPORT NO. 07		IDMS-DC TASK STATISTICS BY TASK CODE R15.0		09/08/99 PAGE 3	
SELECTED FROM: 98001 00:00		TO: 98365 23:59			
ACTUAL: 98300 15:11		END: 98300 15:25			
TASK CODE: ADS2					
3	NUMBER TASK EXECUTIONS				
51	NUMBER PROGRAMS CALLED	6	NUMBER PROGRAMS LOADED		
0	NUMBER TERMINAL READS	2	NUMBER TERMINAL WRITES		
0	NUMBER TERMINAL ERRORS	698	NUMBER GETSTG REQUESTS		
16	NUMBER GETSCR REQUESTS	18	NUMBER PUTSCR REQUESTS		
0	NUMBER DELSCR REQUESTS	0	NUMBER GETQUE REQUESTS		
0	NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS		
171	NUMBER GETTIME REQUESTS	0	NUMBER SETTIME REQUESTS		
1,156	NUMBER DB SERVICE RQSTS	838	NUMBER PAGES READ		
.5431	TASK USER MODE TIME	.5236	TASK SYSTEM MODE TIME		
18.7099	TASK WAIT TIME				
20	NUMBER PAGES WRITTEN	2,285	NUMBER PAGES REQUESTED		
2	NUMBER CALC RECS NO OFLOW	0	NUMBER CALC RECS OFLOW		
120	NUMBER VIA RECS NO OFLOW	0	NUMBER VIA RECS OFLOW		
3,800	NUMBER RECORDS REQUESTED	542	NUMBER RECS CURRENT OF RU		
0	NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED		
1,228	NUMBER CALLS TO DBMS	5,778	TOTAL LOCKS ACQUIRED		
56	NUMBER SR8 SPLITS	44	NUMBER SR8 STORES		
4	NUMBER SR8 SPAWNS	44	NUMBER SR8 ERASES		
18	NUMBER ORPHAN ADOPT	2	NUMBER SR7 STORES		
381	NUMBER BTREE SEARCHES	2	NUMBER SR7 ERASES		
0	NUMBER SQL COMMANDS	0	NUMBER SQL SORTS		
0	NUMBER TUPLES FETCHED	0	NUMBER TUPLES SORTED		
0	NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES		
0	NUMBER ROWS UPDATED				
0	NUMBER ROWS DELETED				

Figure 9-9. Sample SREPORT 007

REPORT NO. 08		IDMS-DC ERUS TASK STATISTICS BY ACCTG DATA - R15.0		09/08/99 PAGE 1	
SELECTED FROM: 98001 00:00		TO: 98365 23:59			
ACTUAL: 98300 15:11		TO: 98300 15:25			
ACCOUNTING DATA: BJENJ002I ;40200000					
1	NUMBER TASK EXECUTIONS				
3	NUMBER PROGRAMS CALLED	1	NUMBER PROGRAMS LOADED		
0	NUMBER TERMINAL READS	0	NUMBER TERMINAL WRITES		
0	NUMBER TERMINAL ERRORS	29	NUMBER GETSTG REQUESTS		
0	NUMBER GETSCR REQUESTS	0	NUMBER PUTSCR REQUESTS		
0	NUMBER DELSCR REQUESTS	0	NUMBER GETQUE REQUESTS		
0	NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS		
328	NUMBER GETTIME REQUESTS	0	NUMBER SETTIME REQUESTS		
10	NUMBER DB SERVICE RQSTS	89	NUMBER PAGES READ		
.0000	TASK USER MODE TIME	.5116	TASK SYSTEM MODE TIME		
9.4807	TASK WAIT TIME				
73	NUMBER PAGES WRITTEN	5,971	NUMBER PAGES REQUESTED		
0	NUMBER CALC RECS NO OFLOW	1	NUMBER CALC RECS OFLOW		
622	NUMBER VIA RECS NO OFLOW	26	NUMBER VIA RECS OFLOW		
5,776	NUMBER RECORDS REQUESTED	660	NUMBER RECS CURRENT OF RU		
16	NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED		
687	NUMBER CALLS TO DBMS	11,680	TOTAL LOCKS ACQUIRED		
0	NUMBER SR8 SPLITS	0	NUMBER SR8 STORES		
0	NUMBER SR8 SPAWNS	0	NUMBER SR8 ERASES		
0	NUMBER ORPHAN ADOPT	0	NUMBER SR7 STORES		
0	NUMBER BTREE SEARCHES	0	NUMBER SR7 ERASES		
0	NUMBER SQL COMMANDS	0	NUMBER SQL SORTS		
0	NUMBER TUPLES FETCHED	0	NUMBER TUPLES SORTED		
0	NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES		
0	NUMBER ROWS UPDATED				
0	NUMBER ROWS DELETED				

Figure 9-10. Sample SREPORT 008

REPORT NO. 09		IDMS-DC ERUS TASK STATISTICS BY PGM NAME - R15.0		09/08/99 PAGE 1	
SELECTED FROM: 98001 00:00		TO: 98365 23:59			
ACTUAL: 98300 15:11		TO: 98300 15:25			
PROGRAM NAME: IDMSDDDL					
1	NUMBER TASK EXECUTIONS				
3	NUMBER PROGRAMS CALLED	1	NUMBER PROGRAMS LOADED		
0	NUMBER TERMINAL READS	0	NUMBER TERMINAL WRITES		
0	NUMBER TERMINAL ERRORS	29	NUMBER GETSTG REQUESTS		
0	NUMBER GETSCR REQUESTS	0	NUMBER PUTSCR REQUESTS		
0	NUMBER DELSCR REQUESTS	0	NUMBER GETQUE REQUESTS		
0	NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS		
328	NUMBER GETTIME REQUESTS	0	NUMBER SETTIME REQUESTS		
10	NUMBER DB SERVICE RQSTS	89	NUMBER PAGES READ		
.0000	TASK USER MODE TIME	.5116	TASK SYSTEM MODE TIME		
9.4807	TASK WAIT TIME				
73	NUMBER PAGES WRITTEN	5,971	NUMBER PAGES REQUESTED		
0	NUMBER CALC RECS NO OFLOW	1	NUMBER CALC RECS OFLOW		
622	NUMBER VIA RECS NO OFLOW	26	NUMBER VIA RECS OFLOW		
5,776	NUMBER RECORDS REQUESTED	660	NUMBER RECS CURRENT OF RU		
16	NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED		
687	NUMBER CALLS TO DBMS	11,680	TOTAL LOCKS ACQUIRED		
0	NUMBER SR8 SPLITS	0	NUMBER SR8 STORES		
0	NUMBER SR8 SPAWNS	0	NUMBER SR8 ERASES		
0	NUMBER ORPHAN ADOPT	0	NUMBER SR7 STORES		
0	NUMBER BTREE SEARCHES	0	NUMBER SR7 ERASES		
0	NUMBER SQL COMMANDS	0	NUMBER SQL SORTS		
0	NUMBER TUPLES FETCHED	0	NUMBER TUPLES SORTED		
0	NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES		
0	NUMBER ROWS UPDATED				
0	NUMBER ROWS DELETED				

Figure 9-11. Sample SREPORT 009

### 9.8.1 Field descriptions

A description of the fields common to SREPORTs 005 through 009 follows:

#### SELECTED FROM/TO

Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

#### ACTUAL/TO

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

#### NUMBER TASK EXECUTIONS

Indicates the total number of tasks executed within the actual time frame of the report. This label appears on SREPORTs 005, 006, 007, 008, and 009.

#### NUMBER OF TRANSACTIONS

Indicates the total number of transactions executed within the actual time frame of the report. This label appears on SREPORTs 010, 011, and 021.

#### NUMBER PROGRAMS CALLED

Indicates the number of programs called.



**NUMBER PROGRAMS LOADED**

Indicates the number of programs loaded from disk (either a load area or a load/core-image library).

**NUMBER TERMINAL READS**

Indicates the number of terminal reads performed.

**NUMBER TERMINAL WRITES**

Indicates the number of terminal writes performed.

**NUMBER TERMINAL ERRORS**

Indicates the number of terminal I/O errors.

**NUMBER GETSTG REQUESTS**

Indicates the number of requests to acquire storage dynamically from storage pool.

**NUMBER GETSCR REQUESTS**

Indicates the number of requests to retrieve scratch records from the DDLDCSCR area.

**NUMBER PUTSCR REQUESTS**

Indicates the number of requests to place scratch records in the the DDLDCSCR area.

**NUMBER DELSCR REQUESTS**

Indicates the number of requests to delete scratch records from the DDLDCSCR area.

**NUMBER GETQUE REQUESTS**

Indicates the number of requests to get queue records from the DDLDCRUN area.

**NUMBER PUTQUE REQUESTS**

Indicates the number of requests to store or replace queue records in the DDLDCRUN area.

**NUMBER DELQUE REQUESTS**

Indicates the number of requests to delete queue records from the DDLDCRUN area.

**NUMBER GETTIME REQUESTS**

Indicates the number of requests for the date and time of day.

**NUMBER SETTIME REQUESTS**

Indicates the number of requests to define an event that is to occur after a specified time interval.

**NUMBER DB SERVICE RQSTS**

Indicates the number of times requests for database services (for example, OBTAIN *record-name*) were issued by the task.

For LRF and SQL programs, this value should be less than or equal to the number of CALLS TO DBMS. You can use these values to evaluate how efficiently the LRF or SQL path extracts data. For example, a program OBTAIN *logical-record* command increments NUMBER DB SERVICE RQSTS by one, but may greatly increment the CALLS TO DBMS value, especially if an area sweep occurs due to the NULL SELECT clause.

#### **NUMBER PAGES READ**

Indicates the number of database pages read from disk.

#### **TASK USER MODE TIME**

Indicates the amount of CPU time (in ten-thousandths seconds) spent executing user code. DC/UCF collects user-mode time statistics if enabled in the STATISTICS TASK clause of the SYSTEM system generation statement.

#### **TASK SYSTEM MODE TIME**

Indicates the amount of CPU time (in ten-thousandths seconds) spent performing DC/UCF services for the task. DC/UCF collects system-mode time statistics if enabled in the STATISTICS TASK clause of the SYSTEM system generation statement.

#### **TASK WAIT TIME**

Indicates the amount of time spent on I/O requests and waiting for other system resources. The value is the difference between wall-clock and CPU time for the task.

#### **NUMBER PAGES WRITTEN**

Indicates the number of database pages physically written to disk for the transaction. A page can be updated several times before it is actually written back to the database.

#### **NUMBER PAGES REQUESTED**

Indicates the number of database pages requested by IDMSDBMS (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool.

Interpretation: The ratio of NUMBER PAGES REQUESTED to NUMBER PAGES READ is the **buffer utilization ratio**. The buffer utilization ratio measures the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio the better. Ratios consistently below 2.0 indicate that processing is random or that the buffer-pool size is too small.

The buffer utilization ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

#### **NUMBER CALC RECS NO OFLOW**

Indicates the number of CALC records stored on the target page.

#### **NUMBER CALC RECS OFLOW**

Indicates the number of CALC records not stored on the target page.

Interpretation: The ratio of NO OVERFLOW records to the total number of CALC records stored is the **CALC cluster ratio**. Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate space utilization is getting high and the database should be tuned.

#### **NUMBER VIA RECS NO OFLOW**

Indicates the number of VIA and/or DIRECT records stored on the target page.

**NUMBER VIA RECS OFLOW**

Indicates the number of VIA and or DIRECT records not stored on the target page.

Interpretation: The ratio of NO OVERFLOW records to the total number of VIA records stored is the **VIA cluster ratio**. Ideally, the ratio should be 1, which indicates no overflow. A value less than 1 or less than the norm indicates very large data clusters, high utilization of space, or small page size.

**NUMBER RECORDS REQUESTED**

Indicates the number of records requested by the DBMS.

Interpretation: The ratio of NUMBER RECORDS REQUESTED to NUMBER PAGES REQUESTED is the **space management ratio**. The space management ratio measures how well space is allocated (for example, VIA options, CALC distribution, and buffering). The higher the ratio the better. Ratios less than 4 or less than the norm indicate the size of the buffer should be increased and database tuning should be performed.

The space management ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

**NUMBER RECORDS CURRENT OF RU**

Indicates the number of records made current of the transaction.

Interpretation: The ratio of NUMBER RECORDS REQUESTED to NUMBER RECORDS CURRENT OR RU is the **effectiveness ratio**. The effectiveness ratio measures how much work the CA-IDMS/DB has to do to find the requested record. The lower the ratio the better. If the ratio is high, examine set options (for example, sorted order or next pointers only) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

**NUMBER FRAGMENTS STORED**

Indicates the number of noncontiguous segments stored for variable-length records.

**NUMBER RECORDS RELOCATED**

Indicates the number of records relocated from the home page.

**NUMBER CALLS TO DBMS**

Indicates the number of calls to the database management system.

**Note:** Execution of each navigational DML request involves one call; execution of each logical record facility (LRF) or SQL request typically involves multiple calls.

**NUMBER SR8 SPLITS**

Indicates the number of SR8 splits.

**NUMBER SR8 SPAWNS**

Indicates the number of SR8 spawns.

**NUMBER ORPHAN ADOPT**

Indicates count of Index members or SR8s whose up-level pointers were corrected to point to the actual SR8 in which they appear.

**NUMBER BTREE SEARCHES**

Indicates number of Btree index probes.

**NUMBER SQL COMMANDS**

Indicates the number of SQL commands executed.

**NUMBER TUPLES FETCHED**

Indicates number of tuples FETCHed.

**NUMBER ROWS INSERTED**

Indicates the number of rows INSERTed.

**NUMBER ROWS UPDATED**

Indicates the number of rows UPDATed.

**NUMBER ROWS DELETED**

Indicates the number of rows DELETED.

**NUMBER TOTAL LOCKS**

Indicates the number of all locks acquired and released by all completed transactions. This is NOT a count of locks currently held.

**NUMBER SR8 STORES**

Indicates count of SR8s created.

**NUMBER SR8 ERASES**

Indicates count of SR8s erased.

**NUMBER SR7 STORES**

Indicates count of SR7s created.

**NUMBER SR7 ERASES**

Indicates count of SR7s erased.

**NUMBER SORTS**

Indicates the number of SQL sorts performed.

**NUMBER TUPLES SORTED**

Indicates the number of rows participating in all sorts.

**NUMBER AM RECOMPILES**

Indicates the number of automatic access module recompilations.

## 9.9 Transaction statistics reports

**Overview:** Transaction statistics record resource usage by transaction (that is, across tasks). At run time, IDMS-DC accumulates transaction statistics when requested to do so by a user program.

The program initiates collection with a BIND TRANSACTION STATISTICS statement and terminates collection with an END TRANSACTION STATISTICS statement. The END TRANSACTION STATISTICS and ACCEPT TRANSACTION STATISTICS statements write statistics to the DC/UCF log file.

►► For more information about transaction statistics and how to collect transaction statistics, see *CA-IDMS System Operations*.

**Summary of transaction statistics reports:** Three statistics reports summarize transaction activity:

SREPORT 010	Summarizes transaction statistics by user identifier.
SREPORT 011	Summarizes transaction statistics by logical terminal identifier.
SREPORT 021	Summarizes transaction statistics by dialog.

REPORT NO. 10		IDMS-DC TRANSACTION STATISTICS BY USER ID R15.0		09/08/99 PAGE 4	
SELECTED FROM: 00000 00:00		TO: 00000 00:00			
ACTUAL: 99249 11:17		TO: 99250 10:51			
USER ID: CMH					
NUMBER PROGRAMS CALLED	404	NUMBER PROGRAMS LOADED	20		
NUMBER TERMINAL READS	22	NUMBER TERMINAL WRITES	22		
NUMBER TERMINAL ERRORS	0	NUMBER GETSBG REQUESTS	1,200		
NUMBER GETSCR REQUESTS	61	NUMBER PUTSCR REQUESTS	71		
NUMBER DELSCR REQUESTS	19	NUMBER GETQUE REQUESTS	48		
NUMBER PUTQUE REQUESTS	33	NUMBER DELQUE REQUESTS	11		
NUMBER GETTIME REQUESTS	313	NUMBER SETTIME REQUESTS	0		
NUMBER DB SERVICE RQSTS	42	NUMBER PAGES READ	180		
TASK USER MODE TIME	.8382	TASK SYSTEM MODE TIME	2.7998		
TASK WAIT TIME	18.2473				
NUMBER PAGES WRITTEN	15	NUMBER PAGES REQUESTED	2,556		
NUMBER CALC RECS NO OFLOW	11	NUMBER CALC RECS OFLOW	0		
NUMBER VIA RECS NO OFLOW	33	NUMBER VIA RECS OFLOW	0		
NUMBER RECORDS REQUESTED	2,737	NUMBER RECORDS CURRENT OF RU	920		
NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED	0		
NUMBER CALLS TO DBMS	581	TOTAL LOCKS ACQUIRED	7,466		
NUMBER SR8 SPLITS	56	NUMBER SR8 STORES	44		
NUMBER SR8 SPAWNS	4	NUMBER SR8 ERASES	44		
NUMBER ORPHAN ADOPT	18	NUMBER SR7 STORES	2		
NUMBER BTREE SEARCHES	381	NUMBER SR7 ERASES	2		
NUMBER SQL COMMANDS	27	NUMBER SQL SORTS	1		
NUMBER TUPLES FETCHED	785	NUMBER TUPLES SORTED	728		
NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES	0		
NUMBER ROWS UPDATED	0				
NUMBER ROWS DELETED	0				

Figure 9-12. Sample SREPORT 010

## 9.9 Transaction statistics reports

REPORT NO. 11		IDMS-DC TRANSACTION STATISTICS BY LTERM ID R15.0		09/08/99 PAGE 13	
SELECTED FROM: 00000 00:00		TO: 00000 00:00			
ACTUAL: 99249 11:17		TO: 99250 10:51			
LTERM ID: LT12022					
NUMBER PROGRAMS CALLED	1,137	NUMBER PROGRAMS LOADED	39		
NUMBER TERMINAL READS	66	NUMBER TERMINAL WRITES	66		
NUMBER TERMINAL ERRORS	0	NUMBER GETSBG REQUESTS	2,351		
NUMBER GETSCR REQUESTS	27	NUMBER PUTSCR REQUESTS	13		
NUMBER DELSCR REQUESTS	21	NUMBER GETQUE REQUESTS	0		
NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS	0		
NUMBER GETTIME REQUESTS	733	NUMBER SETTIME REQUESTS	0		
NUMBER DB SERVICE RQSTS	103	NUMBER PAGES READ	248		
TASK USER MODE TIME	.4968	TASK SYSTEM MODE TIME	3.7788		
TASK WAIT TIME	9.3349				
NUMBER PAGES WRITTEN	0	NUMBER PAGES REQUESTED	1,849		
NUMBER CALC RECS NO OFLOW	0	NUMBER CALC RECS OFLOW	0		
NUMBER VIA RECS NO OFLOW	0	NUMBER VIA RECS OFLOW	0		
NUMBER RECORDS REQUESTED	2,149	NUMBER RECORDS CURRENT OF RU	686		
NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED	0		
NUMBER CALLS TO DBMS	581	TOTAL LOCKS ACQUIRED	7,466		
NUMBER SR8 SPLITS	56	NUMBER SR8 STORES	44		
NUMBER SR8 SPAWNS	4	NUMBER SR8 ERASES	44		
NUMBER ORPHAN ADOPT	18	NUMBER SR7 STORES	2		
NUMBER BTREE SEARCHES	381	NUMBER SR7 ERASES	2		
NUMBER SQL COMMANDS	27	NUMBER SQL SORTS	1		
NUMBER TUPLES FETCHED	785	NUMBER TUPLES SORTED	728		
NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES	0		
NUMBER ROWS UPDATED	0				
NUMBER ROWS DELETED	0				

Figure 9-13. Sample SREPORT 011

REPORT NO. 21		IDMS-DC TRANSACTION STATISTICS BY DIALOG R15.0		09/08/99 PAGE 7	
SELECTED FROM: 00000 00:00		TO: 00000 00:00			
ACTUAL: 99249 15:26		TO: 99250 15:28			
DIALOG: ASFRDEFD					
NUMBER PROGRAMS CALLED	110	NUMBER PROGRAMS LOADED	11		
NUMBER TERMINAL READS	6	NUMBER TERMINAL WRITES	6		
NUMBER TERMINAL ERRORS	0	NUMBER GETSTG REQUESTS	264		
NUMBER GETSCR REQUESTS	18	NUMBER PUTSCR REQUESTS	10		
NUMBER DELSCR REQUESTS	18	NUMBER GETQUE REQUESTS	0		
NUMBER PUTQUE REQUESTS	0	NUMBER DELQUE REQUESTS	0		
NUMBER GETTIME REQUESTS	64	NUMBER SETTIME REQUESTS	0		
NUMBER DATA BASE REQUESTS	0	NUMBER PAGES READ	59		
TASK USER MODE TIME	.0520	TASK SYSTEM MODE TIME	.7729		
TASK WAIT TIME	.1598				
NUMBER PAGES WRITTEN	0	NUMBER PAGES REQUESTED	247		
NUMBER CALC RECS NO OFLOW	0	NUMBER CALC RECS OFLOW	0		
NUMBER VIA RECS NO OFLOW	0	NUMBER VIA RECS OFLOW	0		
NUMBER RECORDS REQUESTED	311	NUMBER RECORDS CURRENT OF RU	112		
NUMBER FRAGMENTS STORED	0	NUMBER RECORDS RELOCATED	0		
NUMBER CALLS TO DBMS	581	TOTAL LOCKS ACQUIRED	7,466		
NUMBER SR8 SPLITS	56	NUMBER SR8 STORES	44		
NUMBER SR8 SPAWNS	4	NUMBER SR8 ERASES	44		
NUMBER ORPHAN ADOPT	18	NUMBER SR7 STORES	2		
NUMBER BTREE SEARCHES	381	NUMBER SR7 ERASES	2		
NUMBER SQL COMMANDS	27	NUMBER SQL SORTS	1		
NUMBER TUPLES FETCHED	785	NUMBER TUPLES SORTED	728		
NUMBER ROWS INSERTED	0	NUMBER AM RECOMPILES	0		
NUMBER ROWS UPDATED	0				
NUMBER ROWS DELETED	0				

Figure 9-14. Sample SREPORT 021

### 9.9.1 Field descriptions

SREPORTs 010, 011, and 021 have fields in common with task and ERUS reports; for a description of each field, see 9.8, “Task and External Request Unit Service (ERUS) statistics reports” on page 9-35 earlier in this chapter.

## 9.10 CA-ADS dialog statistics

**Overview:** If CA-ADS dialog and transaction statistics are enabled, statistics are collected for all or selected dialogs that execute during an application program. DC/UCF collects the following types of dialog statistics each time a dialog issues a control command:

- Statistics for explicitly coded control commands issued by the dialog
- Statistics for implicitly coded control commands issued by the ADS run-time system on behalf of the dialog
- Dialog execution statistics
- Statistics for record buffer block (RBB) usage

Statistics are written to the system log when the number of statistics accumulations equals the checkpoint interval established at system generation and when the application terminates.

►► For more information about collecting these statistics, see the *CA-ADS Reference* and *CA-IDMS System Operations*.

**Note:** Statistics for dialogs executed in batch mode can be collected in a separate log file associated with the batch dialog. For more information, see the *CA-ADS Batch User Guide*.

**Summary of ADS statistics reports:** Three statistics reports summarize dialog run-time activity:

---

SREPORT 018	Summarizes CA-ADS dialog statistics by user.
SREPORT 019	Summarizes CA-ADS dialog statistics by dialog and version number.
SREPORT 020	Summarizes CA-ADS dialog statistics by logical terminal.

---



REPORT NO. 018		ADS STATISTICS BY USER - R15.0		09/08/99	PAGE 1
USER ID : ACM					
DATE :	99249 TIME :	14:07 DIALOG NAME :	ASFASELD	VERSION NUMBER :	1
DATE BIND :	99249 TIME BIND :	14:02 LTERM ID :	LT12002		
DISPLAY COMMAND:	4 DISPLAY CONTINU:	2 INVOKES :		0 LINK TO DIALOGS:	0
LNKS TO PROGRAM:	7 RETURNS :	0 RETURN CONTINUE:		0 TRANSFERS :	0
LEAVE ADS :	1 LEAVE APPLICATN:	0 ABORTS :		0 IMPL DISPLAYS :	0
IMPL INVOKE :	0 IMPL LINK DLGS :	0 IMPL LINK PGMS :		0 IMPL RETURNS :	0
IMPL RET CONT :	0 IMPL TRANSFERS :	1 IMPL LEAVE ADS :		0 IMPL LEAVE PGMS:	0
IMPL ABORTS :	0 PREMAP PROCESS :	4 RESPONSE PROCES:		4 STAT ACCUM CALL:	15
EXPL GET SCRS :	4 EXPL PUT SCRS :	3 EXPL DEL SCRS :		11 WRTE PRINT REQS:	0
PUT NEW DETAILS:	0 PUT CUR DETAILS:	0 GET DETAILS :		0 SIZE OF FDB :	50,688
SIZE OF VDB :	1,228 HIGHEST LNK LEV:	0 LOWEST LNK LEVL:		0 RBB PUT TO SCR :	0
RBB STG HI MARK:	5,144 RBB FREE HI :	3,024 RBB STG LOW MK :		5,144 RBB FREE LOW :	3,024
MOST RBB ACQ :	3,080 LEAST RBB ACQ :	3,080 HICOUNT RBB USE:		2 LOCOUNT RBB USE:	2
DATE :	99249 TIME :	14:07 DIALOG NAME :	ASFASFD	VERSION NUMBER :	1
DATE BIND :	99249 TIME BIND :	14:02 LTERM ID :	LT12002		
DISPLAY COMMAND:	0 DISPLAY CONTINU:	0 INVOKES :		0 LINK TO DIALOGS:	0
LNKS TO PROGRAM:	2 RETURNS :	0 RETURN CONTINUE:		0 TRANSFERS :	0
LEAVE ADS :	0 LEAVE APPLICATN:	0 ABORTS :		0 IMPL DISPLAYS :	0
IMPL INVOKE :	0 IMPL LINK DLGS :	0 IMPL LINK PGMS :		0 IMPL RETURNS :	0
IMPL RET CONT :	0 IMPL TRANSFERS :	1 IMPL LEAVE ADS :		0 IMPL LEAVE PGMS:	0
IMPL ABORTS :	0 PREMAP PROCESS :	1 RESPONSE PROCES:		0 STAT ACCUM CALL:	3
EXPL GET SCRS :	0 EXPL PUT SCRS :	0 EXPL DEL SCRS :		9 WRTE PRINT REQS:	0
PUT NEW DETAILS:	0 PUT CUR DETAILS:	0 GET DETAILS :		0 SIZE OF FDB :	5,272
SIZE OF VDB :	300 HIGHEST LNK LEV:	0 LOWEST LNK LEVL:		0 RBB PUT TO SCR :	0
RBB STG HI MARK:	2,240 RBB FREE HI :	1,844 RBB STG LOW MK :		2,240 RBB FREE LOW :	1,844
MOST RBB ACQ :	200 LEAST RBB ACQ :	200 HICOUNT RBB USE:		1 LOCOUNT RBB USE:	1
**** USER TOTAL ****					
DISPLAY COMMAND:	4 DISPLAY CONTINU:	2 INVOKES :		0 LINK TO DIALOGS:	0
LNKS TO PROGRAM:	9 RETURNS :	0 RETURN CONTINUE:		0 TRANSFERS :	0
LEAVE ADS :	1 LEAVE APPLICATN:	0 ABORTS :		0 IMPL DISPLAYS :	0
IMPL INVOKE :	0 IMPL LINK DLGS :	0 IMPL LINK PGMS :		0 IMPL RETURNS :	0
IMPL RET CONT :	0 IMPL TRANSFERS :	2 IMPL LEAVE ADS :		0 IMPL LEAVE PGMS:	0
IMPL ABORTS :	0 PREMAP PROCESS :	5 RESPONSE PROC :		4 STAT ACCUM CALL:	18
EXPL GET SCRS :	4 EXPL PUT SCRS :	3 EXPL DEL SCRS :		20 WRTE PRINT REQS:	0
PUT NEW DETAILS:	0 PUT CUR DETAILS:	0 GET DETAILS :		0 RECORD COUNT :	2

Figure 9-15. Sample SREPORT 018

## 9.10 CA-ADS dialog statistics

REPORT NO. 019		ADS STATISTICS BY DIALOG AND VERSION NUMBER - R15.0		09/08/99	PAGE 8
DIALOG NAME : ADSOAFNC		VERSION NUMBER: 1			
DATE :	99249	TIME :	09:49	USER ID :	SMT
DATE BIND :	99249	TIME BIND :	09:46	LTERM ID :	LT12011
DISPLAY COMMAND:	21	DISPLAY CONTINU:	21	INVOKES :	3
LNKS TO PROGRAM:	18	RETURNS :	0	RETURN CONTINUE:	0
LEAVE ADS :	0	LEAVE APPLICATN:	0	ABORTS :	0
IMPL INVOKE :	0	IMPL LINK DLGS :	0	IMPL LINK PGMS :	0
IMPL RET CONT :	0	IMPL TRANSFERS :	0	IMPL LEAVE ADS :	0
IMPL ABORTS :	0	PREMAP PROCESS :	42	RESPONSE PROCES:	21
EXPL GET SCRS :	0	EXPL PUT SCRS :	0	EXPL DEL SCRS :	0
PUT NEW DETAILS:	0	PUT CUR DETAILS:	0	GET DETAILS :	0
SIZE OF VDB :	836	HIGHEST LNK LEV:	1	LOWEST LNK LEVL:	1
RBB STG HI MARK:	3,176	RBB FREE HI :	908	RBB STG LOW MK :	3,176
MOST RBB ACQ :	304	LEAST RBB ACQ :	304	HICOUNT RBB USE:	1
**** DIALOG TOTAL ****					
DISPLAY COMMAND:	21	DISPLAY CONTINU:	21	INVOKES :	3
LNKS TO PROGRAM:	18	RETURNS :	0	RETURN CONTINUE:	0
LEAVE ADS :	0	LEAVE APPLICATN:	0	ABORTS :	0
IMPL INVOKE :	0	IMPL LINK DLGS :	0	IMPL LINK PGMS :	0
IMPL RET CONT :	0	IMPL TRANSFERS :	0	IMPL LEAVE ADS :	0
IMPL ABORTS :	0	PREMAP PROCESS :	42	RESPONSE PROC :	21
EXPL GET SCRS :	0	EXPL PUT SCRS :	0	EXPL DEL SCRS :	0
PUT NEW DETAILS:	0	PUT CUR DETAILS:	0	GET DETAILS :	0
				LINK TO DIALOGS:	18
				TRANSFERS :	18
				IMPL DISPLAYS :	0
				IMPL RETURNS :	0
				IMPL LEAVE PGMS:	0
				STAT ACCUM CALL:	99
				WRTE PRINT REQS:	0
				SIZE OF FDB :	23,080
				RBB PUT TO SCR :	0
				RBB FREE LOW :	908
				LOCOUNT RBB USE:	1

Figure 9-16. Sample SREPORT 019

REPORT NO. 020		ADS STATISTICS BY LOGICAL TERMINAL R15.0		09/08/99	PAGE 15
TERMINAL ID : LT12003					
DATE	:	99249	TIME	:	11:22
DATE BIND	:	99249	TIME BIND	:	11:19
DISPLAY COMMAND:	3	DISPLAY CONTINU:	2	INVOKES	:
LNKS TO PROGRAM:	8	RETURNS	:	0	RETURN CONTINUE:
LEAVE ADS	:	1	LEAVE APPLICATN:	0	ABORTS
IMPL INVOKE	:	0	IMPL LINK DLGS	:	0
IMPL RET CONT	:	0	IMPL TRANSFERS	:	1
IMPL ABORTS	:	0	PREMAP PROCESS	:	4
EXPL GET SCRS	:	4	EXPL PUT SCRS	:	3
PUT NEW DETAILS:	0	PUT CUR DETAILS:	0	GET DETAILS	:
SIZE OF VDB	:	1,228	HIGHEST LNK LEV:	0	LOWEST LNK LEVL:
RBB STG HI MARK:	5,144	RBB FREE HI	:	3,024	RBB STG LOW MK
MOST RBB ACQ	:	3,080	LEAST RBB ACQ	:	3,080
				5,144	RBB FREE LOW
				2	LOCOUNT RBB USE
					2
DATE	:	99249	TIME	:	11:22
DATE BIND	:	99249	TIME BIND	:	11:19
DISPLAY COMMAND:	0	DISPLAY CONTINU:	0	INVOKES	:
LNKS TO PROGRAM:	2	RETURNS	:	0	RETURN CONTINUE:
LEAVE ADS	:	0	LEAVE APPLICATN:	0	ABORTS
IMPL INVOKE	:	0	IMPL LINK DLGS	:	0
IMPL RET CONT	:	0	IMPL TRANSFERS	:	1
IMPL ABORTS	:	0	PREMAP PROCESS	:	1
EXPL GET SCRS	:	0	EXPL PUT SCRS	:	0
PUT NEW DETAILS:	0	PUT CUR DETAILS:	0	GET DETAILS	:
SIZE OF VDB	:	300	HIGHEST LNK LEV:	0	LOWEST LNK LEVL:
RBB STG HI MARK:	2,240	RBB FREE HI	:	1,844	RBB STG LOW MK
MOST RBB ACQ	:	200	LEAST RBB ACQ	:	200
				2,240	RBB FREE LOW
				1	LOCOUNT RBB USE
					1
**** LOGICAL TERMINAL TOTAL ****					
DISPLAY COMMAND:	3	DISPLAY CONTINU:	2	INVOKES	:
LNKS TO PROGRAM:	10	RETURNS	:	0	RETURN CONTINUE:
LEAVE ADS	:	1	LEAVE APPLICATN:	0	ABORTS
IMPL INVOKE	:	0	IMPL LINK DLGS	:	0
IMPL RET CONT	:	0	IMPL TRANSFERS	:	2
IMPL ABORTS	:	0	PREMAP PROCESS	:	5
EXPL GET SCRS	:	4	EXPL PUT SCRS	:	3
PUT NEW DETAILS:	0	PUT CUR DETAILS:	0	GET DETAILS	:
				3	STAT ACCUM CALL:
				21	WRITE PRINT REQS:
				0	RECORD COUNT
					2

Figure 9-17. Sample SREPORT 020

## 9.10.1 Field descriptions

A description of the fields common to SREPORTs 018 through 020 follows:

### DATE

Specifies the date the statistics were written to the system log; the date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the Julian day.

### TIME

Specifies the time the statistics were written to the system log.

### DATE BIND

Specifies the date the TSB BIND command was issued; the date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the Julian day.

### TIME BIND

Specifies the time the TSB BIND command was issued.

**DISPLAY COMMAND**

Specifies the number of explicit DISPLAY commands issued by the dialog.

**DISPLAY CONTINU**

Specifies the number of explicit DISPLAY CONTINUE commands issued by the dialog.

**INVOKES**

Specifies the number of explicit INVOKE commands issued by the dialog.

**LINKS TO DIALOGS**

Specifies the number of explicit LINK TO DIALOG commands issued by the dialog.

**LNKS TO PROGRAM**

Specifies the number of explicit LINK TO PROGRAM commands issued by the dialog.

**RETURNS**

Specifies the number of explicit RETURN commands issued by the dialog.

**RETURN CONTINUE**

Specifies the number of explicit RETURN CONTINUE commands issued by the dialog.

**TRANSFERS**

Specifies the number of explicit TRANSFER commands issued by the dialog.

**LEAVE ADS**

Specifies the number of explicit LEAVE ADS commands issued by the dialog.

**LEAVE APPLICATN**

Indicates number of explicit LEAVE APPLICATION commands issued by the dialog.

**ABORTS**

Indicates the number of explicit ABORT commands issued by the dialog.

**IMPL DISPLAYS**

Indicates the number of implicitly generated DISPLAY commands issued during the application.

**IMPL INVOKE**

Specifies the number of implicitly generated INVOKE commands issued during the application.

**IMPL LINK DLGS**

Indicates the number of implicitly generated LINK TO DIALOG commands issued during the application.

**IMPL LINK PGMS**

Indicates the number of implicitly generated LINK TO PROGRAM commands issued during the application.

**IMPL RETURNS**

Indicates the number of implicitly generated RETURN commands issued during the application.

**IMPL RET CONT**

Indicates the number of implicitly generated RETURN CONTINUE commands issued during the application.

**IMPL TRANSFERS**

Indicates the number of implicitly generated TRANSFER commands issued during the application.

**IMPL LEAVE ADS**

Indicates the number of implicitly generated LEAVE ADS commands issued during the application.

**IMPL LEAVE PGMS**

Indicates the number of implicitly generated LEAVE APPLICATION commands issued during the application.

**IMPL ABORTS**

Indicates the number of implicitly generated ABORT commands issued during the application.

**PREMAP PROCESS**

Indicates the number of dialog premap process executions.

**RESPONSE PROCES**

Indicates the number of dialog response process executions.

**STAT ACCUM CALL**

Indicates the number of calls to IDMS-DC to accumulate dialog transaction statistics.

**EXPL GET SCRS**

Specifies the number of explicit dialog requests to retrieve scratch records from the DDLDCSCR area.

**EXPL PUT SCRS**

Specifies the number of explicit dialog requests to place scratch records in the DDLDCSCR area.

**EXPL DEL SCRS**

Specifies the number of explicit dialog requests to delete scratch records from the DDLDCSCR area.

**WRTE PRINT REQS**

Indicates the number of dialog WRITE PRINTER commands.

**PUT NEW DETAILS**

Indicates the number of dialog PUT NEW DETAIL commands.

**PUT CUR DETAILS**

Indicates the number of dialog PUT CURRENT DETAIL commands.

**GET DETAILS**

Indicates the number of dialog GET DETAIL commands.

**SIZE OF FDB**

Specifies the size of the fixed dialog block in bytes. The FDB is the dialog load module generated by the dialog generator.

**SIZE OF VDB**

Specifies the size of the variable dialog block in bytes. The VDB is created dynamically for the issuing dialog at run-time and contains run-time variable information about the dialog.

**HIGHEST LNK LEV**

Indicates the highest level within an application thread at which the dialog was executed. The value for the highest link level is zero.

**LOWEST LNK LEVL**

Indicates the lowest level within an application thread at which the dialog was executed. The higher the value the lower the link level.

**RBB PUT TO SCR**

Indicates the number of record buffer blocks placed as scratch records in the DDLDCSCR area.

**RBB STG HI MARK**

Indicates the most record buffer block storage allocated for all dialogs.

**RBB FREE HI**

Indicates the amount of free record buffer block space when the most storage is allocated for all dialogs.

**RBB STG LOW MK**

Indicates the least record buffer block storage allocated for all dialogs.

**RBB FREE LOW**

Indicates the amount of free space in the record buffer block when the least storage is used.

**MOST RBB ACQ**

Indicates the most space acquired in the record buffer block for the dialog.

**LEAST RBB ACQ**

Indicates the least space acquired in the record buffer block for the dialog.

**HICOUNT RBB USE**

Indicates the highest number of record buffer blocks used.

**LOCOUNT RBB USE**

Indicates the lowest number of record buffer blocks used.

**RECORD COUNT**

(Total summary only) Indicates the total number of dialog statistics records written.

►► For a detailed explanation of the CA-ADS commands, see the *CA-ADS Reference*.

## 9.11 Histogram Report

**Overview:** Histograms show statistical data for events (for example, program loads into the program pool) in terms of frequency of occurrence within predefined value ranges (for example, number of program loads smaller than 250 bytes, between 250 to 500 bytes, and so forth).

►► For a detailed discussion of histogram categories, classes, default values, and the #HSTDEF macro that overrides default values, see *CA-IDMS System Operations*.

**When statistics are written to log:** Like system statistics, histograms are written to the DC/UCF log file at the following times:

- At normal system shutdown
- At the statistics interval established at system generation by the STATISTICS parameter of the SYSTEM statement. The statistics interval can be varied at run time with the DCMT VARY STATISTICS command.
- Upon explicit request by means of a DCMT WRITE STATISTICS command.

**Summary of histogram record subtypes:** SREPORT 001, IDMS Statistics-Histogram Report, produces a histogram for each of 25 histogram record subtypes. Table 9-1 lists each histogram record subtype by identifier and descriptor.

Table 9-1 (Page 1 of 2). Histogram Records

Subtype Identifier	Description
01	PUT JOURNAL request size
02	Program size in bytes
03	Queue record sizes
04	Scratch record sizes
05	User storage size
06	Total size of GET STORAGE requests
07	Programs called by task
08	Programs loaded by task
09	Terminal reads by task
10	Terminal writes by task
11	Terminal errors by task
12	Number of GET STORAGE requests by task
13	Number of GET SCRATCH requests by task

Table 9-1 (Page 2 of 2). Histogram Records

<b>Subtype Identifier</b>	<b>Description</b>
14	Number of PUT SCRATCH requests by task
15	Number of DELETE SCRATCH requests by task
16	Number of GET QUEUE requests by task
17	Number of PUT QUEUE requests by task
18	Number of DELETE QUEUE requests by task
19	Number of GET TIME requests by task
20	Number of SET TIME requests by task
21	Number of database calls by task
22	Number of high stacks by task
23	Total time (in ten-thousandths seconds) spent in user mode by task
24	Total time (in ten-thousandths seconds) spent in system mode by task
25	Total wait time by task

The report below illustrates a page of a sample histogram report; it contains a histogram of the number of programs loaded by task (subtype 8).



REPORT NO. 01				IDMS STATISTICS - HISTOGRAM REPORT R15.0				09/08/99 PAGE 8			
SELECTED FROM: 99249 20:30				TO: 99249 21:50							
ACTUAL: 99249 20:32				TO: 99249 21:07							
		SUB-TYPE	GW-DESCRIPTION		BINS	LOW END	INCREMENT	LOW VALUE	HIGH VALUE	STAT RECORD DATE	STAT RECORD TIME
HISTOGRAM FOR SUB-TYPE		08	PROGRAMS LOADED BY TASK		10	1	2	292	0	99249	21:06
			BELOW	1	292						
		1	TO	2	38						
		3	TO	4	7						
		5	TO	6	3						
		7	TO	8	0						
		9	TO	10	1						
		11	TO	12	0						
		13	TO	14	0						
		15	TO	16	1						
		17	TO	18	0						
		19	TO	20	0						
		21	AND ABOVE...		0						
		TOTAL:			342						

Figure 9-18. Sample SREPORT 001

### 9.11.1 Field descriptions

A description of the fields for the 25 histograms produced by SREPORT 001 follows:

#### SELECTED FROM/TO

Specifies the time period specified on the USE 'SREPORT 000' parameter where **SELECTED FROM** specifies the beginning date and time and **SELECTED TO** specifies the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

#### ACTUAL/TO

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, **ACTUAL** and **TO** represent the time range for the entire input archive file or input file created by SREPORT 099.

#### SUB-TYPE

Specifies the identifier of the histogram record subtype.

#### GW-DESCRIPTION

Specifies the descriptor for the histogram record subtype.

**BINS**

Indicates the number of bins in the histogram; a bin has finite limits. The default is 10.

**LOW END**

Indicates the low end of the range of values of the histogram.

**INCREMENT**

Specifies the increment added to range values for successive bins.

**LOW VALUE**

Indicates the number of occurrences below the value specified for LOW END.

**HIGH VALUE**

Indicates the number of occurrences greater than the high end of the histogram's range of values.

**STAT RECORD DATE**

Specifies the date, in Julian form, the histogram record was written to the DC/UCF log file.

**STAT RECORD TIME**

Specifies the time, in *hh:mm* form, the histogram was written to the DC/UCF log file.

**TOTAL**

Indicates the total of the bin occurrences.

## 9.12 Record Summary Statistics Report

**Overview:** DC/UCF logs 35 types of statistics records to the system log file. Particular statistic reports require specific record types. For example, statistics record 30 provides the statistics summarized in SREPORT 013, IDMS-DC Program Summary.

**Summary of statistics records:** The table below lists each type of statistics record. The record ID numbers correspond to numbers shown in column 1 of SREPORT 017, Summary of Records Read:

Record Number	Statistics Record Type
00	System startup marker record
01/01	Histogram record of PUT JOURNAL DML requests
01/02	Histogram record of program size in pages
01/03	Histogram record of queue record size
01/04	Histogram record of scratch record size
01/05	Histogram record of user storage size
01/06	Histogram record of total requested storage
01/07	Histogram record of programs called by task
01/08	Histogram record of programs loaded by task
01/09	Histogram record of terminal reads by task
01/0A	Histogram record of terminal writes by task
01/0B	Histogram record of terminal errors by task
01/0C	Histogram record of storage requests by task
01/0D	Histogram record of GET SCRATCH requests by task
01/0E	Histogram record of PUT SCRATCH requests by task
01/0F	Histogram record of DELETE SCRATCH requests by task
01/10	Histogram record of GET QUEUE requests by task
01/11	Histogram record of PUT QUEUE requests by task
01/12	Histogram record of DELETE QUEUE requests by task
01/13	Histogram record of GET TIME requests by task
01/14	Histogram record of SET TIME requests by task
01/15	Histogram record of database calls by task

<b>Record Number</b>	<b>Statistics Record Type</b>
01/16	Histogram record of high stacks by task
01/17	Histogram record of user time by task
01/18	Histogram record of system time by task
01/19	Histogram record of wait time by task
01/1A	Histogram record of high RCE count
01/1B	Histogram record of high RLE count
01/1C	Histogram record of high DPE count
07/01	Histogram record of response time by line
01	System-wide statistics record
02	Task statistics record
03	Transaction statistics record
04	Task code statistics record
05	Program statistics record
06	Queue statistics record
07	Line statistics record
08	Physical terminal statistics record
09	CA-ADS dialog statistics record

### 9.12.1 Contents of SREPORT 017

SREPORT 017, the Summary of Records Read report, itemizes the number of each type of statistics record written to the log file within a defined period of time. Systems administrators can use SREPORT 017 to determine which statistics reports can be run based upon available statistics records.

REPORT NO. 17		SUMMARY OF RECORDS READ R15.0		09/08/99 PAGE 1	
SELECTED FROM: 00000 00:00		TO: 00000 00:00			
ACTUAL: 99249 17:03		TO: 99250 08:18			
STATISTICS RECORD TYPES STL/HST	BELOW DATE/TIME RANGE	BEFORE TRIGGER STARTUP RECORDS	PASSED TO STATISTICS ROUTINES	DATES/TIMES OF 1ST AND LAST RECS OF EACH TYPE PASSED TO THE ROUTINES	
00 - START UP	0	0	1	99250 7:48	99250 7:48
01/01 - PUTJRN REQ SIZE	0	0	6	99249 17:13	99250 8:18
01/02 - LOADED PGM SIZE	0	0	6	99249 17:13	99250 8:18
01/03 - QUEUE RECORD SIZE	0	0	6	99249 17:13	99250 8:18
01/04 - SCRATCH REC SIZE	0	0	6	99249 17:13	99250 8:18
01/05 - USER STG SIZE	0	0	6	99249 17:13	99250 8:18
01/06 - TOTAL GETSTG SIZE	0	0	6	99249 17:13	99250 8:18
01/07 - PGMS CALLED	0	0	0		
01/08 - PGMS LOADED	0	0	0		
01/09 - TERMINAL READS	0	0	0		
01/0A - TERMINAL WRITES	0	0	0		
01/0B - TERMINAL ERRORS	0	0	0		
01/0C - GET STORAGES	0	0	0		
01/0D - GET SCRATCHES	0	0	0		
01/0E - PUT SCRATCHES	0	0	0		
01/0F - DEL SCRATCHES	0	0	0		
01/10 - GET QUEUE	0	0	0		
01/11 - PUT QUEUE	0	0	0		
01/12 - DEL QUEUE	0	0	0		
01/13 - GET TIMES	0	0	0		
01/14 - SET TIMES	0	0	0		
01/15 - DB CALLS	0	0	0		
01/16 - STACK HIWATER	0	0	0		
01/17 - USER-MODE TIME	0	0	0		
01/18 - SYSTEM-MODE TIME	0	0	0		
01/19 - WAIT TIME	0	0	0		
01/1A - HIGH RCE COUNT	0	0	0		
01/1B - HIGH RLE COUNT	0	0	0		
01/1C - HIGH DPE COUNT	0	0	0		
07/01 - LINE RESPONSE	0	0	40	99249 17:13	99250 18:50
01 - SYSTEM STATS	0	0	6	99249 17:13	99250 18:50
02 - TASK STATS	0	0	115	99249 17:13	99250 18:50
03 - TRANSACTION STATS	0	0	0		
04 - TASK CODE STATS	0	0	376	99249 17:13	99250 8:18
05 - PROGRAM STATS	0	0	5,903	99249 17:13	99250 8:18
06 - QUEUE STATS	0	0	6	99249 17:13	99250 8:18
07 - LINE STATS	0	0	44	99249 17:13	99250 8:18
08 - PTERM STATS	0	0	199	99249 17:13	99250 8:18
09 - ADS STATS	0	0	0		
TOTALS	0	0	6,726		

Figure 9-19. Sample SREPORT 017

## 9.12.2 Field descriptions

A description of the fields in the Summary of Records Read report follows:

### SELECTED FROM/TO

Gives the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM represents the beginning date and time and SELECTED TO represents the ending date and time. The date is in Julian form, *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day. The time is in *hh:mm* form, where *hh* is hours based on a 24-hour clock and *mm* is minutes.

**ACTUAL/TO**

Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

**COUNTS FOR STATISTICS RECORDS**

Specifies the type of statistic record. The table above lists each type of statistics record, as defined in DSECT #STLDS; for more information, see the *CA-IDMS DSECT Reference*.

**BELOW DATE/TIME RANGE**

Indicates the number of records of each type logged to the log file before the date and time specified in the SELECTED FROM field.

**BEFORE TRIGGER STARTUP RECORDS**

Indicates the number of records of each type logged to the log file after the date and time specified in the SELECTED FROM field, but before the system startup marker record (type 00) corresponding to the session indicator specified on the USE 'SREPORT 000' parameter.

**PASSED TO STATISTICS ROUTINES**

Indicates the number of records processed by statistics routines.

**DATES/TIMES OF 1ST AND LAST RECS OF EACH TYPE PASSED TO THE ROUTINES**

Specifies the date and time of the first record of each type passed to a statistics routine and the date and time of the last record passed to a statistics routine. If no records were passed to a routine, the field is blank.

## Chapter 10. Modifying CA-IDMS Reports

---

10.1 Common modifications . . . . .	10-3
10.2 Steps to modify reports . . . . .	10-4
10.3 Obtaining report module listings . . . . .	10-5
10.3.1 Types of listings . . . . .	10-5
10.4 Coding in CA-CULPRIT . . . . .	10-6
10.4.1 Overview of CA-CULPRIT . . . . .	10-6
10.4.2 CULPRIT coding guidelines . . . . .	10-6
10.4.3 Types of CA-CULPRIT parameters . . . . .	10-6
10.4.3.1 Parameters within each category . . . . .	10-6
10.4.4 Report-specific and global parameters . . . . .	10-8
10.4.5 Edit parameters . . . . .	10-8
10.4.5.1 Functions of each column . . . . .	10-8
10.4.6 Summary of parameter types . . . . .	10-9
10.5 Temporary modifications . . . . .	10-12
10.5.1 Copying the report module . . . . .	10-12
10.5.2 Deleting columns . . . . .	10-12
10.5.3 Changing headings . . . . .	10-13
10.5.4 Selecting specific data . . . . .	10-14
10.5.5 Specifying a new sort sequence . . . . .	10-14
10.5.6 Moving columns . . . . .	10-14
10.6 Permanent modifications . . . . .	10-16





## 10.1 Common modifications

CA-IDMS reports can be modified, either temporarily or permanently, to suit site-specific needs. Some common modifications users make are:

- Reformatting a report (for example, sorting the data differently, changing column sequence, reducing the number of characters per line)
- Changing report content (for example, adding or deleting columns, rewording headings, modifying process operations)
- Selecting specific information for a report (for example, reporting on all occurrences after a certain date)

## 10.2 Steps to modify reports

To modify report output, proceed with the following steps:

1. Obtain a sample of the report to be modified.
2. Obtain a listing of the report module.
3. Plan the modifications.

For modifying D-, C-, and AREPORTs, see the *Data Dictionary Network Diagram* and &U\$DDR. to determine which records to select and which elements in those records are needed.

For modifying J- and SREPORTs, see the record layouts described in Chapter 8, "CA-IDMS/DB Journal Reports — JREPORTs and Chapter 9, "DC/UCF Statistics Reports — SREPORTs."

4. Code the modifications.
5. Obtain a sample of the modified report.

The rest of this chapter explains how to carry out steps 2 and 4.

## 10.3 Obtaining report module listings

### 10.3.1 Types of listings

The user can obtain a printed listing, an online listing, or a listing output to punched cards or a disk file.

*Printed listing:* To obtain a printed listing, use Method 1 or Method 2 below:

- **Method 1** — Include PARAM=LIST with the report-specific control statements submitted to run the report. PARAM=LIST lists the CA-CULPRIT parameters in the Sequential and Input Parameter Listings associated with report output.
- **Method 2** — Run DREPORT 076, Module Key Report, using the DICTNAME and DBNAME options on the DATABASE parameter to name the dictionary that stores the report modules, and specifying the name of the report module in the KEY parameter.

In the example below, DREPORT 008 is requested; the report module is stored in the CULPDICT dictionary:

```
DATABASE DICTNAME=CULPDICT DBNAME=CULPDICT
DREPORT=076
KEY MOD-NAME-067 'DREPORT 008'
```

*Online listing:* To obtain an online listing, sign on to the Integrated Data Dictionary (IDD) and submit the DISPLAY MODULE statement to the DDDL compiler, as shown in the example below:

```
DISPLAY MODULE 'DREPORT 008' AS SYNTAX.
```

*Output to cards or disk:* To output the report module to punched cards or a disk file, run DREPORTs 051 or 052, respectively. An example appears below. For more information about submitting these reports, see 2.11, “Special-purpose report modules” on page 2-76.

```
DATABASE DICTNAME=CULPDICT
DREPORT=051
KEY MOD-NAME-067 'DREPORT 008'
```

## 10.4 Coding in CA-CULPRIT

### 10.4.1 Overview of CA-CULPRIT

Because all CA-IDMS report modules are coded using CA-CULPRIT parameters, some knowledge of CA-CULPRIT is required to modify the reports. This is a brief introduction to CA-CULPRIT. For more information about CA-CULPRIT, see the following manuals:

- *CA-CULPRIT Reference*
- *CA-CULPRIT User Guide*

### 10.4.2 CULPRIT coding guidelines

CA-CULPRIT parameters are coded in an 80-column format. These general rules apply to all CA-CULPRIT parameters:

- All parameters must be coded in uppercase letters.
- Column 1 is always blank, except on a continuation line or a USE parameter. An asterisk (\*) in column 1 designates a continuation line.
- CA-CULPRIT parameters must not extend past column 72.
- Comments are introduced by a dollar (\$) sign.
- The USE parameter cannot appear in any run that includes REPORT=, =MACRO, or =COPY parameters.

### 10.4.3 Types of CA-CULPRIT parameters

There are five major categories of CA-CULPRIT parameters:

- Input definition parameters
- Output definition parameters
- Process definition parameters
- Work field definition parameters
- CA-IDMS/DB retrieval parameters

#### 10.4.3.1 Parameters within each category

Each category has a function and includes several types of parameters, as follows:

- **Input definition parameters** define the source of input for the run:

---

INPUT	Defines the physical characteristics of the input file or identifies the subschema for the database to be accessed. A REPORT= request automatically generates an INPUT parameter. Runs that call report modules with the USE parameter must supply an INPUT parameter.
REC	Defines the location, length, and data type of fields that appear in the CA-CULPRIT code. REC parameters are generated automatically for CA-IDMS report modules.
SELECT or BYPASS	Defines criteria by which records are selected for processing.

---

- **Output definition parameters** define the format of the report output:

---

OUTPUT	Defines the physical characteristics of the output; for example, the number of characters per line. By default, output is a printed report with 132 characters per line and 55 lines per page.
SORT	Defines the order in which records are output.
3	Identifies a title parameter. A title parameter outputs a title at the top of each page.
4	Identifies a heading parameter. Headings are output under the title line, if any, at the top of each page.
5	Identifies a detail line.
6	Identifies a total line.

---

- **Process definition parameters** define report-specific process operations, such as arithmetic and data comparisons:

---

7	Identifies a process parameter applied to input data.
8	Identifies a process parameter applied to data in CA-CULPRIT's output phase of processing.

---

- **Work field definition parameters** define work fields required for report processing. Identifiers for work field parameters are **0**, **1**, and **15**.
- **CA-IDMS/DB retrieval parameters** define methods to access CA-IDMS/DB database records:

---

PATH	Identifies record retrieval routes through the database.
KEY	Identifies a key field by which to access database records directly, rather than by an area sweep.

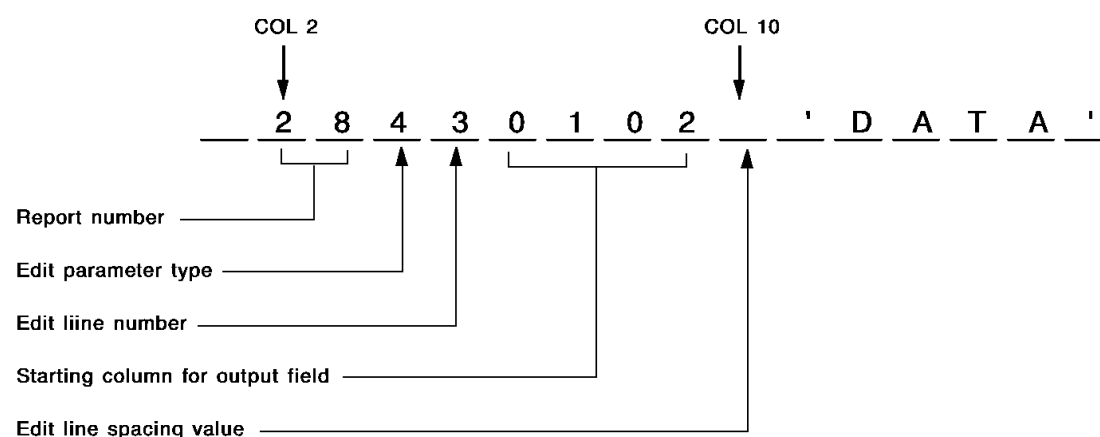
---

### 10.4.4 Report-specific and global parameters

Output and process definition parameters are report-specific. A 2-digit report number, starting in column 2, must precede each parameter identifier. Work field definition parameters are either report-specific or global; GW must appear in columns 2 through 3 for a global work field.

### 10.4.5 Edit parameters

Parameters 4, 5, and 6 are collectively known as edit parameters. Edit parameters define where a field is output in a report. The figure below illustrates the fixed format portion of an edit parameter.



#### 10.4.5.1 Functions of each column

The columns in the figure above function as follows:

- **Columns 2 and 3** identify the 2-digit report number.
- **Column 4** identifies the type of edit line: a heading, detail, or total line.
- **Column 5** identifies a particular definition of an edit line. Each type of edit line can have up to eight different definitions. For example, the code shown in the figure below specifies six heading lines for DREPORT 028.
- **Columns 6 through 9** identify the location of the output field on the edit line. The location can be **absolute**, with a value in the range 0001 through 0132 for a printed report, or **relative**, with a value in the range \*000 to \*999.
- **Column 10** identifies spacing associated with the edit lines. Possible values for printed output are:
  - Blank — no blank lines
  - 0 — one blank line
  - Hyphen — two blank lines

### 10.4.6 Summary of parameter types

The table below summarizes the CA-CULPRIT parameters and indicates what each can do in modifying CA-IDMS reports.

Table 10-1. CA-CULPRIT Parameter Types

Identifier	Function	Modifications
INPUT	Defines input	Modify the CA-CULPRIT buffer size or subschema name
REC	Defines fields	Modify the name, start position, length, or data type of a field
SELECT or BYPASS	Selects records	Select only records that meet certain criteria
OUTPUT	Defines output	Modify the report's line size or number of lines per page
SORT	Sorts	Modify the order in which the report presents information
3	Outputs a title	Add/delete/change the report title
4	Outputs a heading	Add/delete/change report headings
5	Outputs a detail line	Add/delete/change columns of information
6	Outputs a total line	For summary reports, add/delete/change columns of information
7	Defines input processing	Modify or add processing applied to input data
8	Defines total processing	For summary reports, modify or add processing applied to total-time data
0, 1, or 15	Defines work fields	Add if required for additional processing
PATH	Accesses database records	Modify or add to access different database records
KEY	Accesses specified database records	Add to access particular database records

Below are the CA-CULPRIT parameters that produced the sample report below:

## 10.4 Coding in CA-CULPRIT

```
PATHU1 00AK-012,S-010,SS-026,SSR-032,RCDACT-059,PROG-051
280 FUNC-TBL.13 'CONNECT' 'OBTAIN'
* 'DISCONNECT' 'FIND'
* 'STORE' 'ERASE'
* 'CURRENCY ACCEPTED' 'MODIFY'
* 'GET' 'BIND'
* 'FIND KEEP' 'OBTAIN KEEP'
* ' ' '$ FUNCTIONS'
280 KEY-TBL.12 7 43 11 3 12 2 15 8 5 14 23 63 $ KEY TO FUNCTION TBL
280OUTPUT D LP=51
285SORT S-NAME-010 SS-NAM-026 SSR-NAM-032
* PROG-NAME-051 PROG-VER-051 FUNC-TBL.DISP STAR-HD
283 DATA DICTIONARY REPORTER
28410001 'DREPORT 028'
28410053 'IDMS RECORD ACTIVITY REPORT'
28420001-STAR-HD
28420045 STAR-HD
28420089 STAR-HD
28430102 '---- D A T E ----'
28440001 'SCHEMA'
28440010 'VER'
28440016 'SUBSCHEMA'
28440027 'RECORD'
28440060 'PROGRAM'
28440069 'VER'
28440075 'USAGE'
28440095 'TIMES'
28440102 'COMPILED'
28440112 'CREATED'
28450001 STAR-HD
28450045 STAR-HD
28450089 STAR-HD
28460001 ' '
28510001 S-NAM-010
28510009 S-SER-010 FM 'ZZZZZ9'
28510016 SS-NAM-026
28510027 SSR-NAM-032
28510060 PROG-NAME-051
28510068 PROG-VER-051 FM 'ZZZZZ9'
28510075 FUNC-TBL.DISP
28510095 RA-COUNT-059 FM 'ZZZZZ9'
28510102 PROG-DATE-051
28510111 DATE-CREATED-051
287010 IF PATH-ID NE 'U1' DROP $ DROP N/A PATHS
287020 IF S-NAM-010 EQ 'NON IDMS' DROP $ NON IDMS FILES ONLY
287100 $ TABLE LOOKUP TO DETERMINE SUBSCRIPT FOR FUNCTION TABLE
287 MOVE RA-FUNCT-059 TO IND $ KEY FOR LOOKUP
287110 CALL US62 (KEY-TBL 'N' 8 12 'S' IND DISP)
287120 IF DISP NE 0 150 $ IF FOUND, TAKE
287 MOVE 13 TO DISP $ ELSE SET TO BLANKS
287150 TAKE
```

Figure 10-1. Sample CULPRIT source code and output



REPORT NO. 28			DATA DICTIONARY REPORTER				09/08/99 PAGE 1	
DREPORT 028			IDMS RECORD ACTIVITY REPORT					
*****								
						---- D A T E ----		
SCHEMA	VER	SUBSCHEMA	RECORD	PROGRAM	VER	USAGE	TIMES	COMPILED CREATED
*****								
EMPSCHM	100	EMPSS01	COVERAGE	EMPINQ	1	BIND	1	01/07/99 01/07/99
EMPSCHM	100	EMPSS01	DENTAL-CLAIM	EMPINQ	1	BIND	1	01/07/99 01/07/99
EMPSCHM	100	EMPSS01	DEPARTMENT	D3	1	BIND	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	DEPARTMENT	D4	1	BIND	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	DEPARTMENT	D6	1	BIND	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	DEPARTMENT	D6	1	OBTAIN	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	DEPARTMENT	EMPINQ	1	BIND	1	01/07/99 01/07/99
EMPSCHM	100	EMPSS01	DEPARTMENT	TESTPROG	1	BIND	1	02/12/99
EMPSCHM	100	EMPSS01	DEPARTMENT	TESTPROG	1	OBTAIN	2	02/12/99
EMPSCHM	100	EMPSS01	EMPLOYEE	D3	1	BIND	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	EMPLOYEE	D3	1	FIND	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	EMPLOYEE	D3	1	MODIFY	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	EMPLOYEE	D4	1	BIND	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	EMPLOYEE	D4	1	ERASE	1	02/11/99 02/11/99
EMPSCHM	100	EMPSS01	EMPLOYEE	D4	1	FIND	1	02/11/99 02/11/99

Figure 10-2. Sample report

## 10.5 Temporary modifications

**Run-time modifications:** When a report module is modified at run time, the modifications apply only to the run for which they are specified. You can make the following types of run-time modifications:

- Delete or change the parameters that define the report modules with a CA-CULPRIT USE parameter.
- Add new CA-CULPRIT parameters after the REPORT= request parameter or in conjunction with the USE parameter.

### 10.5.1 Copying the report module

**USE parameter:** To make temporary changes or deletions to the code of a report module, a USE parameter is required. A USE parameter replaces the REPORT= parameter in the user-supplied code. Additionally, the following parameters must be supplied:

- An INPUT parameter that defines the subschema of the dictionary to be accessed (D-, C-, and AREPORTs only); INPUT parameters are always required for J- and SREPORTs.
- A report module that performs initial processing and field definitions (D- and JREPORTs only)

**USE parameter clauses:** USE parameter clauses perform several functions. The three most useful clauses are:

- **DROP**, which drops parameters
- **KEEP**, which retains parameters and drops any that are not specified
- **CHANGE**, which modifies parameters

**Example:** The code below copies in DREPORT modules 000 and 028. The INPUT parameter allocates a 10,000-byte buffer and accesses information defined by subschema IDMSNWKA.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
INPUT 10000 DB SS=IDMSNWKA
USE 'DREPORT 000'
USE 'DREPORT 028'
```

### 10.5.2 Deleting columns

**DROP clause of the USE parameter:** The code below deletes the SCHEMA, VER, and SUBSCHEMA columns shown in the sample report above. Each character string begins with 5, which designates an edit parameter for detail lines. The 1 that follows designates the edit line number. The remaining four numbers designate the absolute column position for each field.

```

DROP '510001' AND $DROP SCHEMA
      '510009' AND $DROP VER
      '510016'    $DROP SUBSCHEMA

```

The figure below shows DREPORT 038 modified by this code:

REPORT NO. 28		DATA DICTIONARY REPORTER				09/08/99 PAGE 1	
DREPORT 028		IDMS RECORD ACTIVITY REPORT					
*****							
		---- D A T E ----					
SCHEMA	VER	SUBSCHEMA	RECORD	PROGRAM	VER	USAGE	TIMES COMPILED CREATED
*****							
			COVERAGE	EMPINQ	1	BIND	1 01/07/99 01/07/99
			DENTAL-CLAIM	EMPINQ	1	BIND	1 01/07/99 01/07/99
			DEPARTMENT	D3	1	BIND	1 02/11/99 02/11/99
			DEPARTMENT	D4	1	BIND	1 02/11/99 02/11/99
			DEPARTMENT	D6	1	BIND	1 02/11/99 02/11/99
			DEPARTMENT	D6	1	OBTAIN	1 02/11/99 02/11/99
			DEPARTMENT	EMPINQ	1	BIND	1 01/07/99 01/07/99
			DEPARTMENT	TESTPROG	1	BIND	1 02/12/99
			DEPARTMENT	TESTPROG	1	OBTAIN	2 02/12/99
			EMPLOYEE	D3	1	BIND	1 02/11/99 02/11/99
			EMPLOYEE	D3	1	FIND	1 02/11/99 02/11/99

### 10.5.3 Changing headings

**CHANGE clause of the USE parameter:** In the code shown below, the headings for the columns just deleted are modified as follows:

- The first CHANGE clause modifies the literals SCHEMA, VER, and SUBSCHEMA to indicate a specific schema and subschema. Note that double quotation marks enclose the literals, which include single quotation marks.
- The second CHANGE clause changes the edit line number for the headings from 4 to 3.

```

CHANGE "'SCHEMA'" TO "'EMPSCHM VERSION 100'" AND
      "'SUBSCHEMA'" TO "'SUBSCHEMA EMPSS01'"
CHANGE '440001' TO '430001' AND
      '440016' TO '430023'
DROP   '440010'                                $DROP SCHEMA VERSION HEADING

```

The figure below shows DREPORT 028 as it appears after these changes in the code.

REPORT NO. 28		DATA DICTIONARY REPORTER				09/08/99 PAGE 1	
DREPORT 028		IDMS RECORD ACTIVITY REPORT					
*****							
EMPSCHM VERSION 100		SUBSCHEMA EMPSS01		---- D A T E ----			
RECORD		PROGRAM		VER	USAGE	TIMES	COMPILED CREATED
*****							
COVERAGE		EMPINQ		1	BIND	1	01/07/99 01/07/99
DENTAL-CLAIM		EMPINQ		1	BIND	1	01/07/99 01/07/99
DEPARTMENT		D3		1	BIND	1	02/11/99 02/11/99
DEPARTMENT		D4		1	BIND	1	02/11/99 02/11/99
DEPARTMENT		D6		1	BIND	1	02/11/99 02/11/99
DEPARTMENT		D6		1	OBTAIN	1	02/11/99 02/11/99

### 10.5.4 Selecting specific data

To agree with the headings modified above, the code must select those records associated with subschema EMPSS01 and version 100 of EMPSCHM. The process parameter shown below specifies sequence number 030; CA-CULPRIT positions this parameter between existing process parameters with sequence numbers 020 and 100.

```
287030 IF S-NAM-010 NE 'EMPSCHM' AND
*       S-SER-010 NE 100 AND
*       SS-NAM-026 NE 'EMPSCHM'      DROP
```

### 10.5.5 Specifying a new sort sequence

The code below replaces the existing SORT parameter with a new SORT parameter. The new SORT parameter orders the report contents by verb usage within program name.

```
DROP SORT
-
-
28SORT PROG-NAME-051 PROG-VER-051 FUNC-TBL.DISP STAR-HD
```

The figure below shows DREPORT 028 with a new sort sequence:

REPORT NO. 28 DREPORT 028	DATA DICTIONARY REPORTER IDMS RECORD ACTIVITY REPORT	09/08/99 PAGE 1
*****		
EMPSCHM VERSION 100	SUBSCHEMA EMPSS01	----
RECORD	PROGRAM VER USAGE	DATE
*****		
OFFICE	D3	1 BIND
EMPLOYEE	D3	1 BIND
DEPARTMENT	D3	1 BIND
EMPLOYEE	D3	1 FIND
EMPLOYEE	D3	1 MODIFY
OFFICE	D4	1 BIND
EMPLOYEE	D4	1 BIND
DEPARTMENT	D4	1 BIND
EMPLOYEE	D4	1 ERASE
EMPLOYEE	D4	1 FIND
OFFICE	D6	1 BIND
EMPLOYEE	D6	1 BIND
DEPARTMENT	D6	1 BIND

### 10.5.6 Moving columns

**Use CHANGE clause of the USE parameter:** Since the report contents are sorted by program name, the program name should appear in the left-most column. The code shown below rearranges the columns and column headings so that they appear as in the report output below.

```

-
-
CHANGE '0060' TO '0001' AND $MOVE PROGRAM TO COLUMN 1
      '0068' TO '0010' AND $MOVE VER TO COLUMN 10
      '0069' TO '0010' AND
      '0075' TO '0018' AND $MOVE USAGE TO COLUMN 18
      '0095' TO '0038' AND $MOVE TIMES TO COLUMN 38
      '0027' TO '0048' AND $MOVE RECORD TO COLUMN 48
      '0102' TO '0085' AND $MOVE DATE COMPILED TO COLUMN 85
      '0111' TO '0095' AND $MOVE DATE CREATED TO COLUMN 95
      '0112' TO '0095'

```

REPORT NO. 28  
DREPORT 028

DATA DICTIONARY REPORTER  
IDMS RECORD ACTIVITY REPORT

09/08/99 PAGE 1

```

*****
EMPSCHM VERSION 100  SUBSCHEMA EMPSS01          ---- D A T E ----
PROGRAM  VER    USAGE          TIMES    RECORD          COMPILED  CREATED
*****

```

D3	1	BIND	1	OFFICE	02/11/99	02/11/99
D3	1	BIND	1	EMPLOYEE	02/11/99	02/11/99
D3	1	BIND	1	DEPARTMENT	02/11/99	02/11/99
D3	1	FIND	1	EMPLOYEE	02/11/99	02/11/99
D3	1	MODIFY	1	EMPLOYEE	02/11/99	02/11/99
D4	1	BIND	1	OFFICE	02/11/99	02/11/99
D4	1	BIND	1	EMPLOYEE	02/11/99	02/11/99
D4	1	BIND	1	DEPARTMENT	02/11/99	02/11/99
D4	1	ERASE	1	EMPLOYEE	02/11/99	02/11/99
D4	1	FIND	1	EMPLOYEE	02/11/99	02/11/99
D6	1	BIND	1	OFFICE	02/11/99	02/11/99
D6	1	BIND	1	EMPLOYEE	02/11/99	02/11/99

## 10.6 Permanent modifications

**Modify dictionary module:** To permanently modify an existing module in the data dictionary, use one of the methods shown below:

*Method 1:* Use the DDDL compiler online in command mode:

- Display the module to be modified:  
DISPLAY MODULE NAME IS 'JREPORT 003' AS SYNTAX.
- Change the module name in the ADD MODULE statement; for example, JREPORT 103.
- Use full screen editing to modify the report.
- Press ENTER to store the new module in the dictionary.

*Method 2:* Use the DDDL compiler online in menu mode, as shown in the following series of screens:

```

                                COMPUTER ASSOCIATES INTERNATIONAL
                                *** MASTER SELECTION ***
IDD REL 15.0                                TOP
→

      DICTIONARY NAME...:DOCUDICT      NODE NAME...:

      USER NAME.....:lhn
      PASSWORD.....:

      USAGE MODE.....:X UPDATE      _ RETRIEVAL

      PFKEY SIMULATION...:X OFF      _ ON

_ ATTR = ATTRIBUTE      <PF2>      _ PROC = PROCESS      <PF3>
_ CLAS = CLASS          <PF4>      _ PROG = PROGRAM      <PF5>
_ ELEM = ELEMENT        <PF6>      _ RECD = RECORD      <PF7>
_ FILE = FILE           <PF8>      _ TABL = TABLE      <PF9>
x MODU = MODULE         <PF10>     _ USER = USER       <PF11>
_ ENTL = USER DEFINED ENTITY LIST  _ SYST = SYSTEM
_ MSGS = MESSAGE
_ QFIL = QFILE
_ DISP = DISPLAY ALL      _ OPTI = OPTIONS
                          _ HELP = HELP      <PF1>

```

```

      IDD REL 15.0                      *** MODULE ENTITY ***                      MODU
      →
                                     DICT=DOCUDICT

      DISPLAY      MODULE NAME.....:jreport 403
      _ MODIFY
      x ADD        VERSION NUMBER...:1      _ HIGHEST  _ NEXT HIGHEST
      _ DELETE     _ LOWEST  _ NEXT LOWEST

                                     LANGUAGE.....:
                                     TEXT.....:

                                     DESCRIPTION.....:

      _ SRCE = MODULE SOURCE      <PF9>      _ MODX = MODULE EXTENSION <PF11>
      _ MSYS = WITHIN SYSTEM
      _ REGN = USER REGISTRATION <PF2>      _ PUBL = PUBLIC ACCESS      <PF3>
      _ CLAT = CLASS/ATTRIBUTES <PF4>      _ RKEY = RELATIONAL KEYS    <PF5>
      _ COMM = COMMENTS          <PF6>      _ COML = COMMENT KEY LIST <PF7>
      _ HIST = HISTORY           <PF8>      x COPY = SAME AS/COPY FROM
      _ XREF = CROSS REFERENCE   <PF10>     _ HELP = HELP              <PF1>

```

```

      IDD REL 15.0                      *** MODULE COPY ***                      COPY
      →
                                     MODULE 'JREPORT 403' VERSION 1

      SAME AS MODULE NAME.....:
      VERSION NUMBER...:      _ HIGHEST  _ LOWEST
      LANGUAGE.....:

      COPY FROM MODULE NAME...: jreport 003
      VERSION NUMBER...:      _ HIGHEST  _ LOWEST
      LANGUAGE.....:
          x SOURCE TEXT
          _ MODULES
          _ SYSTEMS
          _ ATTRIBUTES
          _ USERS
          _ ALL COMMENT TYPES
          _ COMMENTS
          _ DEFINITION
          _ USER DEFINED COMMENT (COMMENT KEY)
          _ USER DEFINED NEST (RELATIONAL KEY)

```

```

      IDD REL 15.0                *** MODULE ENTITY ***                MODU
      →
                                MODULE 'JREPORT 403' VERSION 1 DISPLAYED

      _ DISPLAY      MODULE NAME.....:JREPORT 403
      x MODIFY
      _ ADD          VERSION NUMBER...:1      _ HIGHEST      _ NEXT HIGHEST
      _ DELETE                                _ LOWEST       _ NEXT LOWEST

                                LANGUAGE.....:
                                TEXT.....:

                                DESCRIPTION.....:

      x SRCE = MODULE SOURCE      <PF9>      _ MODX = MODULE EXTENSION <PF11>
      _ MSYS = WITHIN SYSTEM
      _ REGN = USER REGISTRATION <PF2>      _ PUBL = PUBLIC ACCESS      <PF3>
      _ CLAT = CLASS/ATTRIBUTES <PF4>      _ RKEY = RELATIONAL KEYS   <PF5>
      _ COMM = COMMENTS          <PF6>      _ COML = COMMENT KEY LIST <PF7>
      _ HIST = HISTORY            <PF8>      _ COPY = SAME AS/COPY FROM <PF1>
      _ XREF = CROSS REFERENCE    <PF10>     _ HELP = HELP

```

```

      IDD REL 15.0                *** MODULE SOURCE ***                SRCE
      →                                PAGE 1 LINE 1                1/22
                                MODULE 'JREPORT 403' VERSION 1

      ---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---
      03$00**** 'J' REPORTS IDMS JOURNAL FILE 09/08/99 ROUTINE-JRPT003
      030 WK-ABRT-FLAG '12345678901' $ FLAG ABNORMALLY ENDED RUN UNITS
      030 WK-CNTR 1 $ AUTOMATIC COUNTER
      030OUTPUT D
      03SORT PROGRAM-NAME TRANSACT-ID DDR-HD
      03410001 'REPORT NO. 03'
      03410051 DDR-HD
      03410106 DATE
      03410122 'PAGE'
      03410127 PAGE
      03420001 'JREPORT 003'
      03420055 'PROGRAM I/O STATISTICS'
      034200460 '----- PAGES -----'
      0351*010 PROGRAM-NAME HH 'PROGRAM'
      0351*020 TRANSACT-ID FM 'ZZZZZ9999' HH 'TRANSACTION ID'
      0351*030 WK-ABRT-FLAG
      0351*040 READ FM 'ZZZZZZZZ' HH ' READ'
      0351*050 WRITTEN FM 'ZZZZZZZZ' HH ' WRITTEN'
      0351*060 PAGE-REQUESTS FM 'ZZZZZZZZ' HH 'REQUESTED'
      0351*070 '
      037010 IF TYPE NE ('ENDJ' 'ABRT') DROP
      037015 IF TYPE EQ 'ABRT' 030
      037020 MOVE ' ' TO WK-ABRT-FLAG
      037025 TAKE
      037030 MOVE '** ABORT **' TO WK-ABRT-FLAG
      037035 TAKE
                                * * * END OF DATA * * *

```

**Method 3:** Copy the report module to cards using DREPORT 051, as follows:

```

REPORT=051
KEY MOD-NAME-067 'JREPORT 003'

```



Add, delete, or replace cards as required, using the modified cards as input to the DDDL compiler.

*Method 4:* Copy the report to a disk file using DREPORT 052, as follows:

```
REPORT=52  
KEY MOD-NAME-067 'JREPORT 003'
```

Edit the file as necessary, using the edited module as input to the DDDL compiler.

**Retrieve module using REPORT= or USE parameters:** Once the modified module is stored in the data dictionary, it can be retrieved by specifying its module number on a REPORT= parameter or a USE parameter. A good rule of thumb is to base the name of the modified module on the existing module name; for example, DREPORT *xyy*, where *x* is a digit in the range 1 through 9 and *yy* represents the last two digits of the original module number.



# Chapter 11. Other CA-IDMS Reporting Facilities

---

- 11.1 Overview . . . . . 11-3
- 11.2 Online reporting facilities . . . . . 11-4
  - 11.2.1 CA-OLQ . . . . . 11-4
  - 11.2.2 IDMS-DC/UCF OnLine PLOG (OLP) . . . . . 11-4
  - 11.2.3 DC/UCF DCMT DISPLAY commands . . . . . 11-5
  - 11.2.4 IDMS-DC/UCF OPER WATCH commands . . . . . 11-5
  - 11.2.5 IDD DDDL DISPLAY commands . . . . . 11-5
  - 11.2.6 CA-IDMS Performance Monitor . . . . . 11-6
- 11.3 Batch reporting facilities . . . . . 11-7



## 11.1 Overview

In addition to the information supplied by CA-IDMS reports, other CA-IDMS-supplied facilities generate reports or online displays that provide data dictionary definitions and run-time statistics. This chapter describes these online and batch reporting facilities.

## 11.2 Online reporting facilities

Users can display information online with the following tools:

- CA-OLQ
- IDMS-DC/UCF OnLine PLOG
- IDMS-DC/UCF DCMT commands
- IDMS-DC/UCF OPER commands
- IDD DDDL DISPLAY commands
- CA-IDMS Performance Monitor

Each item is discussed separately below.

### 11.2.1 CA-OLQ

**Generate reports on dictionary data:** CA-OLQ can be used to generate reports on information contained in a DC/UCF dictionary. The screen below shows an example of an CA-OLQ-generated report:

```

                                CURRENT REPORT
MAP-NAME-098 PREP-BY-098 PROG-NAME-051 DATE-CREATED-051

EMPMPAP      MAPR10.0      EMPINQ      11/10/99
JKDMPAP1     JKD           D6           02/11/99
EMPJOB       DEH           EMP0BTD     01/22/99
JKDTEST1     JKD           JKDDIAL1    11/19/99
JKDTEST1     JKD           JKDDIAL    11/19/99
JKDMPAP      JKD           D4           02/11/99
JKDMPAP      JKD           D2           02/11/99
      END OF REPORT - PAGE          1
```

CA-OLQ reports supplement information supplied by dictionary (DREPORT), IDMS-DC/UCF (CREPORT), and CA-ADS (AREPORT) reports.

►► For more information about CA-OLQ, see the *CA-OLQ Reference*.

### 11.2.2 IDMS-DC/UCF OnLine PLOG (OLP)

**Displays current contents of the log area:** OLP displays the current contents of the DDLDCLOG area of the dictionary. OLP can be used to display system messages, system trace information, and snap dumps.

The screen below shows information displayed by OLP.

```

FROM    ON    TO    ON    COL PRT SKIP LOG TYPES          ROLL STATUS
08:40:50 1999-09-19 08:55:50 1999-09-19 010 OFF 2345 (WT/TR/DU/ ) 040

08:42 IDMS DC258003 V15 USER KYM SIGNED ON LTERM LT12002 AT 08:42:49.64 86.09
08:42 IDMS DC402009 V15 DICTNAME CORPDICT HAS BEEN SET
08:44 IDMS DC258003 V15 USER BXP SIGNED ON LTERM LT12003 AT 08:44:56.13 86.09
08:44 IDMS DC402009 V15 DICTNAME MIDADICT HAS BEEN SET
08:44 IDMS DC258003 V15 USER ALP SIGNED ON LTERM LT12004 AT 08:44:58.75 86.09
08:44 IDMS DC402009 V15 DICTNAME CORPDICT HAS BEEN SET
08:45 IDMS DC402009 V15 PRINT CLASS HAS BEEN SET
08:45 IDMS DC259001 V15 USER KYM SIGNED OFF LTERM LT12002 AT 08:45:27.70 86.0
08:45 IDMS DC259001 V15 USER ALP SIGNED OFF LTERM LT12004 AT 08:45:38.21 86.0
08:48 IDMS DC075102 V15 T5 REQUESTING TERM PRT12051(CT013007) LINE VTAM12
08:49 IDMS DC258003 V15 USER LHN SIGNED ON LTERM LT12002 AT 08:49:54.77 86.09
08:49 IDMS DC402009 V15 DICTNAME DOCUDICT HAS BEEN SET
08:49 IDMS DC258003 V15 USER RPM SIGNED ON LTERM LT12004 AT 08:49:58.86 86.09
08:49 IDMS DC402009 V15 DICTNAME CORPDICT HAS BEEN SET
08:50 IDMS DC258003 V15 USER GAF SIGNED ON LTERM LT12005 AT 08:50:03.54 86.09
08:50 IDMS DC402009 V15 PRINT CLASS HAS BEEN SET

```

►► For more information about this reporting tool, see *CA-IDMS System Operations*.

### 11.2.3 DC/UCF DCMT DISPLAY commands

**Display definitions and run-time statistics:** DC/UCF DCMT DISPLAY commands display definitions and run-time statistics associated with DC/UCF systems.

DCMT commands complement information supplied by IDMS-DC/UCF system (CREPORT) and statistics (SREPORT) reports.

►► For more information about DCMT commands, see *CA-IDMS System Tasks and Operator Commands*.

### 11.2.4 IDMS-DC/UCF OPER WATCH commands

**Display dynamic system run-time statistics:** IDMS-DC/UCF OPER WATCH commands display dynamic system run-time statistics associated with IDMS-DC/UCF systems.

OPER WATCH commands complement information supplied by IDMS-DC/UCF system (CREPORT) and statistics (SREPORT) reports.

►► For more information about OPER WATCH commands, see *CA-IDMS System Tasks and Operator Commands*.

### 11.2.5 IDD DDDL DISPLAY commands

**Display entity occurrences defined to dictionary:** DDDL can be used in command mode or menu mode to display entity occurrences defined to the data dictionary. The screen below shows the contents displayed by submitting the following command to IDD:

```
DISPLAY MAP NAME IS SYBMAP AS SYNTAX.
```

```
DISPLAY MAP NAME IS SYBMAP AS SYNTAX.  
ADD  
MAP NAME IS SYBMAP VERSION IS 1  
WITHIN PANEL SYBMAP-OLMPANEL VERSION IS 1  
DATE CREATED IS      03/14/99  
DATE LAST UPDATED IS 03/14/99  
PREPARED BY SYB  
REVISED  BY SYB  
PUBLIC ACCESS IS ALLOWED FOR ALL  
DATE LAST COMPILED IS 03/14/99  
TIME LAST COMPILED IS 153322  
MAP RELEASE I.D. IS R2  
NUMBER OF NON LITERAL FIELDS IS 1  
NUMBER OF RECORDS IN MAP IS 1  
PFLD NAME TO SET CURSOR AT IS OLMPF-0003  
ASSOCIATED WITH RECORD SYBREC VERSION 1  
.
```

Information display using DDDL in command mode or menu mode complements information supplied by data dictionary (DREPORT), IDMS-DC/UCF system (CREPORT), and CA-ADS (AREPORT) reports.

►► For more information on this reporting tool, see the *IDD DDDL Reference*.

### 11.2.6 CA-IDMS Performance Monitor

**Display system statistics:** The CA-IDMS Performance Monitor is a performance and tuning tool you can use to monitor hardware and software utilization in a DC/UCF system.

**Three components:** The Performance Monitor consists of three components:

- Realtime monitor — Displays specific system-resource statistics at the time of the request
- Interval monitor — Displays system-wide, wait-time statistics for a unit of time
- Application monitor — Displays statistics about resource usage by individual program

►► For more information about the online component of CA-IDMS Performance Monitor, see the *CA-IDMS Performance Monitor User Guide*.



## 11.3 Batch reporting facilities

You can display information with the following batch reporting tools:

- CA-IDMS/DB utilities
- CA-IDMS Performance Monitor reports

The utilities are described below. A brief overview of the CA-IDMS Performance Monitor appears earlier in this chapter. For more information about the batch component of the Performance Monitor, see *CA-IDMS Performance Monitor System Administration*.

**Summary of utilities reports:** **Utilities** are batch facilities that perform system services and frequently generate report output. The table below lists CA-IDMS-supplied utilities that supplement CA-IDMS reports.

Utility	Information in Report Output
ADSORPTS(1)	CA-ADS dialog and application statistics: <ul style="list-style-type: none"> <li>■ Dialog reports — summary dialog information, process module source statements, dialog-associated records, contents of fixed dialog block (FDB)</li> <li>■ Application reports — application task codes, global records, functions, responses</li> </ul>
ADSOBPLG	Run-time CA-ADS/Batch application and dialog statistics (also called CA-ADS/Batch Print Log Utility in the <i>ADS/Batch Features Guide</i> )
IDMSDBAN(2)	Database structure statistics: <ul style="list-style-type: none"> <li>■ Area statistics</li> <li>■ Set statistics</li> <li>■ Record statistics</li> <li>■ Set analysis data for four types of sets</li> </ul>

Utility	Information in Report Output
IDMSRPTS(2)	<p>Data dictionary definitions for schemas, subschemas, physical databases, and nonschema created entities:</p> <ul style="list-style-type: none"> <li>■ Schema reports — areas, files, subschemas, record types, and sets defined in a specific schema</li> <li>■ Subschema reports — record types, sets, areas, logical records, including paths and program activity, in a specified subschema</li> <li>■ Schema-independent reports — module, IDD-built record, protocol, user, and q-file definitions</li> <li>■ Physical database definition reports — segment and DMCL listings; physical database definitions</li> </ul>
PRINT INDEX(2)	Structure of system-owned indexes and indexed sets
PRINT JOURNAL(2)	<p>Journal file statistics</p> <ul style="list-style-type: none"> <li>■ Status of disk journal files</li> <li>■ Journal page utilization</li> </ul>
PRINT LOG(2)	System statistics logged to DDLCLOG area of the data dictionary
PRINT PAGE(2)	Database page content in hexadecimal/decimal format
PRINT SPACE(2)	Space utilization in one or more areas or segments
RHDCMPUT	Definition, screen image, and source code of map and panel occurrences (also called the Mapping Batch Utility in the IDMS-DC/UCF Mapping Facility)
ROLLBACK and ROLLFORWARD(2)	<p>Journal record statistics:</p> <ul style="list-style-type: none"> <li>■ Checkpoints and ENDJ checkpoint statistics for each transaction</li> <li>■ Before and after database record images in hexadecimal/decimal format</li> </ul>
(1) See <i>CA-ADS Reference</i>	
(2) See <i>CA-IDMS Utilities</i>	

# Appendix A. OS/390 JCL

---

- A.1 Running in local mode . . . . . A-3
  - A.1.1 Central version modifications . . . . . A-6



## A.1 Running in local mode

JCL for executing CA-IDMS reports in **local mode** is shown below.

### CULPRIT (OS/390)

```
//CULPRIT EXEC PGM=CULPRIT,REGION=1024K
//STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR
// DD DSN=idms.loadlib,DISP=SHR
//SORTLIB DD DSN=sys1.sortlib,DISP=SHR
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SORTPRNT DD SYSOUT=A
//SORTMSG DD SYSOUT=A
//SYS004 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYS005 DD DSN=&&SPRMWORK,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
//SYS006 DD DSN=&&SEXTWORK,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=VB,LRECL=2044,BLKSIZE=4628)
//SYS007 DD DSN=&&SRTPWORK,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(TRK,(1,1)),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYS008 DD DSN=&&NSRTWORK,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=VB,LRECL=512,BLKSIZE=4628)
//SORTWK01 DD DSN=&&WRKAWORK,
// UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK02 DD DSN=&&WRKBWORK,
// UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK03 DD DSN=&&WRKCWORK,
// UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK04 DD DSN=&&WRKDWORK,
// UNIT=disk,SPACE=(CYL,(5,2))
//CULSRT11 DD *
SORT FIELDS=(1,69,A),FORMAT=BI
RECORD TYPE=F,LENGTH=(320,,320)
/*
//SYSIN4 DD DUMMY,DCB=BLKSIZE=80
//VSAMCTRL DD DUMMY
//CULLIB DD DSN=idms.srclib,DISP=SHR
//SYS002 DD DSN=user.keyfile,DISP=SHR
//SYS010 DD DSN=user.inputfil,DISP=OLD,
// UNIT=tape,VOL=SER=vvvvvv
```

```

//SYS020 DD DSN=user.nonprint,DISP=(NEW,CATLG),
//          UNIT=tape,VOL=SER=vvvvvv,
//          DCB=(DSORG=PS,LRECL=1111,BLKSIZE=bbbb)
//dictdb  DD DSN=idms.appldict.ddldml,DISP=SHR
//dloddb  DD DSN=idms.appldict.ddldclod,DISP=SHR
//dccat   DD DSN=idms.appldict.ddlcat,DISP=SHR
//dcatx   DD DSN=idms.appldict.ddlcatx,DISP=SHR
//dcatl   DD DSN=idms.appldict.ddlcatlod,DISP=SHR
//dirldb  DD DSN=idms.sysdir1.ddldml,DISP=SHR
//dcmmsg  DD DSN=idms.sysmsg.ddldcmmsg,DISP=SHR
//asfdml  DD DSN=idms.asfdict.ddldml,DISP=SHR
//asflod  DD DSN=idms.asfdict.ddldclod,DISP=SHR
//asfdefn DD DSN=idms.asfdict.asfdefn,DISP=SHR
//asfdata DD DSN=idms.asfdict.asfdata,DISP=SHR
//sysjrn1 DD DUMMY
//syspch  DD SYSOUT=B,DCB=BLKSIZE=80
/*
//SYSIDMS DD *
DMCL=dmcl-name
Other SYSIDMS parameters, as appropriate
/*
//SYSIN   DD *
CULPRIT parameters
/*
//*

```

appldict	name of the application dictionary
asfdata	ddname of the ASF data (IDMSR-AREA2) area; required to execute IREPORTs
ASFDEFN	ddname of the ASF data definition (IDMSR-AREA) area; required to execute IREPORTs
asfdml	ddname of the ASF dictionary definition (DDLDMML) area; required to execute IREPORTs
asflod	ddname of the ASF dictionary definition load (DDLDCLOD) area; required to execute IREPORTs
bbbb	blocksize of the file
dcmmsg	ddname of the system message (DDLDCMSG) area
dictdb	ddname of the application dictionary definition area
dirldb	ddname of the IDMSDIRL definition (DDLDMML) area; to execute A, C, and DREPORTs, use a dictionary built by IDMSDIRL
disk	symbolic device name for work files
dloddb	ddname of the application dictionary definition load area; required for CREPORT 053
dccat	ddname of the SQL application dictionary catalog (DDLDCAT) area
ddcatx	ddname of the SQL application dictionary catalog index (DDLDCATX) area

---

ddcatl	ddname of the SQL application dictionary catalog load (DDLCLATL) area
dmcl-name	name of the DMCL used by the central version
idms.appldict.ddldclod	data set name of the application dictionary definition load (DDLDCLOD) area; required for CREPORT 053
idms.appldict.ddldml	data set name of the application dictionary definition (DDLDM) area
idms.asfdict.asfdata	data set name of the ASF data (IDMSR-AREA2) area; required for IREPORTs
idms.asfdict.asfdefn	data set name of the ASF data definition (IDMSR-AREA) area; required for IREPORTs
idms.asfdict.ddldclod	data set name of the ASF data definition load (DDLDCLOD) area; required for IREPORTs
idms.asfdict.ddldml	data set name of the ASF dictionary definition (DDLDM) area; required for IREPORTs
idms.dba.loadlib	data set name of the load library containing the DMCL and database name table load modules
idms.loadlib	data set name of the load library containing the CA-IDMS executable modules
idms.srclib	data set name of PDS containing parameters to be copied (necessary only if USE, =COPY, or =MACRO features are being used)
idms.srclib(sort1)	data set name and member name of stored sort parameters, as established during installation
idms.sysdirl.ddldml	data set name of the IDMSDIRL definition (DDLDM) area
idms.sysmsg.ddldcmsg	data set name of the system message (DDLDCMSG) area
sysdirl	name of the IDMSDIRL dictionary
sysjrnl	ddname of the tape journal file
syspch	punched card output (required for DREPORT 051 and CREPORT 051, omitted otherwise)
sys1.sortlib	data set name of system sort library
tape	symbolic device name for tape file
user.inputfil	data set name for primary input file (required for JREPORT and SREPORT runs)
user.keyfile	data set name of a key file (necessary only if a keyfile is input)

---

---

user.nonprint	data set name for nonprint/nonpunch output (required for SREPORT 099, DREPORT 052, CREPORT 052; omitted otherwise)
vvvvvv	volume serial number

---

**Note:** If the user's subschema is not in the CA-IDMS/DB load library, then the load library that contains the module must also be included in the STEPLIB concatenation when running in local mode.

## A.1.1 Central version modifications

To run CA-IDMS reports under **central version**:

1. Remove the following DD statements:

```
//dictdb DD DSN=idms.appldict.ddldml,DISP=SHR
//dloddb DD DSN=idms.appldict.ddldclod,DISP=SHR
//dirldb DD DSN=idms.sysdir1.ddldml,DISP=SHR
//asfdml DD DSN=idms.asfdict.ddldml,DISP=SHR
//asflod DD DSN=idms.asfdict.ddldclod,DISP=SHR
//ASFDEFN DD DSN=idms.asfdict.asfdefn,DISP=SHR
//asfdata DD DSN=idms.asfdict.asfdata,DISP=SHR
//sysjrn1 DD DUMMY
```

2. Add the following DD statement for the system control file anywhere after STEPLIB:

```
//sysctl DD DSN=idms.sysctl,DISP=SHR
```

---

sysctl	ddname of the SYSCTL file
idms.sysctl	data set name of SYSCTL file

---



# Appendix B. VSE/ESA JCL

---

- B.1 Running in local mode . . . . . B-3
  - B.1.1 Central version modifications . . . . . B-4
- B.2 IDMSLBLS procedure . . . . . B-5
  - B.2.1 What is the IDMSLBLS procedure? . . . . . B-5
- B.3 IDMSLBLS procedure listing . . . . . B-6



## B.1 Running in local mode

JCL to CA-IDMS reports in **local mode** appears below:

### CULPRIT (VSE/ESA)

```
// JOB      CULPRIT
// DLBL      idms150,'idms150.library'
// EXTENT    SYSnnn,nnnnnn,,,ssss,1111
// ASSGN     SYSnnn,DISK,VOL=nnnnnn,SHR
// LIBDEF    *,SEARCH=CA-IDMS 15.0 libraries
// EXEC      PROC=IDMSLBLS
// OPTION     LOG,CATAL,DUMP
// ASSGN     SYS009,IGN
// ASSGN     SYS001,nnnnnn,,,ssss,1111
// DLBL      SORTWK1,'WORK',0
// EXTENT     SYS001,nnnnnn,,,ssss,1111
// ASSGN     SYS004,X'ppp'
// ASSGN     SYS005,X'ddd'
// DLBL      SYS005,'PARMS',0
// EXTENT     SYS005,nnnnnn,,,ssss,1111
// ASSGN     SYS006,X'ddd'
// DLBL      SYS006,'EXTRACT',0
// EXTENT     SYS006,nnnnnn,,,ssss,1111
// ASSGN     SYS007,X'ddd'
// DLBL      SYS007,'SORTCARD',0
// EXTENT     SYS007,nnnnnn,,,ssss,1111
// ASSGN     SYS008,X'ddd'
// DLBL      SYS008,'NOSORT',0
// EXTENT     SYS008,nnnnnn,,,ssss,1111
// ASSGN     SYS002,X'ttt'
// TLBL      SYS002,'user.keyfile'
// ASSGN     SYS010,X'ttt'
// TLBL      SYS010,'user.inputfil'
// ASSGN     SYSPCH,X'ccc'
// ASSGN     SYS020,X'ttt'
// TLBL      SYS020,'user.nonprint',15
// EXEC      CULPRIT
CULPRIT parameters
/*
user input file, if on cards
/*
optional restart parameter
/*
/&
```

<u>idms150</u>	dtfname of the CA-IDMS library
<u>'idms150.library'</u>	data set name of CA-IDMS libraries, as established during installation
<u>SYSnnn</u>	Logical unit of the volume for which the extent is effective
ccc	device assignment (channel and unit) for punched output (required for DREPORT 051 and CREPORT 051)

---

culprit.srclib	file-id of system library that contains parameters to be copied (ASSGN necessary if USE, =COPY, or =MACRO is used; DLBL and EXTENT necessary if the parameters are maintained on a private library)
ddd	device assignment (channel and unit) for disk files
IDMSLBLS	Name of the procedure provided at installation that contains the file definitions for CA-IDMS dictionary, database, disk journal, and SYSIDMS file definition  ►► For a complete listing of IDMSLBLS, see B.2, “IDMSLBLS procedure” on page B-5 later in this section
llll	number of tracks assigned for file
nnn	
nnnnnn	serial number for disk storage device
ppp	device assignment (channel and unit) for printed output
ssss	starting track of file
SYS009	logical unit of the CA-IDMS tape journal file
SYS020	logical unit for first tape or disk output file
ttt	device assignment (channel and unit) for tape files (files may use disk instead of tape, in which case a device assignment, DLBL and EXTENT information are also required)
nnnnnn	volume serial number
user.inputfil	file-id of the input file (required for JREPORTS and SREPORTS)
user.keyfile	file-id of the key file (necessary only if a key file is input)
user.nonprint,15	file-id and retention period for nonprint/nonpunch output (required for SREPORT 099, DREPORT 052, and CREPORT 052; otherwise omitted)

---

## B.1.1 Central version modifications

To run CA-IDMS reports under the **central version**, add a SYSCTL file to your job control; for example:

```
// ASSGN  SYSnnn,DISK,VOL=vvvvvv,SHR
// DLBL   SYSCTL,'idms.sysctl',2099/365,SD
// EXTENT SYSnnn,nnnnnn,,ssss,2
```

## B.2 IDMSLBLS procedure

### B.2.1 What is the IDMSLBLS procedure?

IDMSLBLS is a procedure provided during a CA-IDMS VSE/ESA installation. It contains file definitions for the CA-IDMS components listed below. These components are provided during installation:

- Dictionaries
- Sample databases
- Disk journal files
- SYSIDMS file

Tailor the IDMSLBLS procedure to reflect the filenames and definitions in use at your site and include this procedure in VSE/ESA JCL job streams.

The sample VSE/ESA JCL provided in this document includes the IDMSLBLS procedure. Therefore, individual file definitions for CA-IDMS dictionaries, sample databases, disk journal files, and SYSIDMS file are not included in the sample JCL.

## B.3 IDMSLBLS procedure listing

```

* ----- LIBDEFS -----
// LIBDEF *,SEARCH=idmslib.sublib
// LIBDEF *,CATALOG=user.sublib
/* ----- LABELS -----
// DLBL idmslib, 'idms.library', 2099/365
// EXTENT ,nnnnnn,,ssss,1500
// DLBL dccat, 'idms.system.dccat', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,31
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dccatl, 'idms.system.dccatlod', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dccatx, 'idms.system.dccatx', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,11
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcdml, 'idms.system.ddldml', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcloclod, 'idms.system.ddldcloclod', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,21
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dclog, 'idms.system.ddldclog', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,401
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcrun, 'idms.system.ddldcrun', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,68
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcscr, 'idms.system.ddldcscr', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,135
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcmsg, 'idms.sysmsg.ddldcmsg', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dclocscr, 'idms.sysloc.ddlocscr', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dirldb, 'idms.sysdirl.ddldml', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dirllod, 'idms.sysdirl.ddldcloclod', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,2
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL empdemo, 'idms.empdemo1', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,11
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL insdemo, 'idms.insdemo1', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL orgdemo, 'idms.orgdemo1', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL empldem, 'idms.sqldemo.empldemo', 2099/365, DA
// EXTENT SYSnnn,nnnnnn,,ssss,11

```

```

// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL infodem,'idms.sqldemo.infodem',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL projdem,'idms.projseg.projdemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL indxdem,'idms.sqldemo.indxdemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL sysctl,'idms.sysctl',2099/365,SD
// EXTENT SYSnnn,nnnnnn,,ssss,2
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL secdd,'idms.sysuser.ddlsec',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dictdb,'idms.appldict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dloddb,'idms.appldict.ddldclod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL sqldd,'idms.syssql.ddlcat',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL sqllod,'idms.syssql.ddlcatl',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL sqlxdd,'idms.syssql.ddlcatx',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL asfdml,'idms.asfdict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL asflod,'idms.asfdict.asflod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,401
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL asfdata,'idms.asfdict.asfdata',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL ASFDEFN,'idms.asfdict.asfdefn',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL j1jrn1,'idms.j1jrn1',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,54
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL j2jrn1,'idms.j2jrn1',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,54
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL j3jrn1,'idms.j3jrn1',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,54
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL SYSIDMS,'sysidms.parms'
// EXTENT SYSnnn,nnnnnn,,ssss,ttt
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
/+
/*

```

---

idmslib.sublib

Name of the sublibrary within the library containing  
CA-IDMS modules

---

---

<u>user.sublib</u>	Name of the sublibrary within the library containing user modules
<u>idmslib</u>	Filename of the file containing CA-IDMS modules
<u>idms.library</u>	File-ID associated with the file containing CA-IDMS modules
<u>SYSnnn</u>	Logical unit of the volume for which the extent is effective
<u>nnnnnn</u>	Volume serial identifier of appropriate disk volume
<u>ssss</u>	Starting track (CKD) or block (FBA) of disk extent
dccat	Filename of the system dictionary catalog (DDL CAT) area
idms.system.dccat	File-ID of the system dictionary catalog (DDL CAT) area
dccatl	Filename of the system dictionary catalog load (DDL CATLOD) area
idms.system.dccatlod	File-ID of the system dictionary catalog load (DDL CATLOD) area
dccatx	Filename of the system dictionary catalog index (DDL CATX) area
idms.system.dccatx	File-ID of the system dictionary catalog index (DDL CATX) area
dcdml	Filename of the system dictionary definition (DDL DML) area
idms.system.ddldml	File-ID of the system dictionary definition (DDL DML) area
dclod	Filename of the system dictionary definition load (DDL DCLOD) area
idms.system.ddldclod	File-ID of the system dictionary definition load (DDL DCLOD) area
dclog	Filename of the system log area (DDL DCLOG) area
idms.system.ddldclog	File-ID of the system log (DDL DCLOG) area
dcrun	Filename of the system queue (DDL DCRUN) area
idms.system.ddldcrun	File-ID of the system queue (DDL DCRUN) area
dcscr	Filename of the system scratch (DDL DCSCR) area
idms.system.ddldcscr	File-ID of the system scratch (DDL DCSCR) area
dcmsg	Filename of the system message (DDL DCMSG) area

---



---

idms.sysmsg.ddldcmsg	File-ID of the system message (DDLDCMSG) area
dclscr	Filename of the local mode system scratch (DDLOCSCR) area
idms.sysloc.ddlocscr	File-ID of the local mode system scratch (DDLOCSCR) area
dirlldb	Filename of the IDMSDIRL definition (DDLDDL) area
idms.sysdirl.ddldml	File-ID of the IDMSDIRL definition (DDLDDL) area
dirlld	Filename of the IDMSDIRL definition load (DDLDCLOD) area
idms.sysdirl.dirlld	File-ID of the IDMSDIRL definition load (DDLDCLOD) area
empdemo	Filename of the EMPDEMO area
idms.empdemo1	File-ID of the EMPDEMO area
insdemo	Filename of the INSDEMO area
idms.insdemo1	File-ID of the INSDEMO area
orgdemo	Filename of the ORGDEMO area
idms.orgdemo1	File-ID of the ORGDEMO area
empldem	Filename of the EMPLDEMO area
idms.sqldemo.empldemo	File-ID of the EMPLDEMO area
infodem	Filename of the INFODEMO area
idms.sqldemo.infodemo	File-ID of the INFODEMO area
projdem	Filename of the PROJDEMO area
idms.projseg.projdemo	File-ID of the PROJDEMO area
indxdem	Filename of the INDXDEMO area
idms.sqldemo.indxdemo	File-ID of the INDXDEMO area
sysctl	Filename of the SYSCTL file
idms.sysctl	File-ID of the SYSCTL file
secdd	Filename of the system user catalog (DDLSEC) area
idms.sysuser.ddlsec	File-ID of the system user catalog (DDLSEC) area
dictdb	Filename of the application dictionary definition area
idms.appldict.ddldml	File-ID of the application dictionary definition (DDLDDL) area
dloddb	Filename of the application dictionary definition load area

---

---

idms.appldict.ddldclod	File-ID of the application dictionary definition load (DDLDCLOD) area
sqldd	Filename of the SQL catalog (DDLCAT) area
idms.syssql.ddlcat	File-ID of the SQL catalog (DDLCAT) area
sqllod	Filename of the SQL catalog load (DDLCATL) area
idms.syssql.ddlcatl	File-ID of SQL catalog load (DDLCATL) area
sqlxdd	Filename of the SQL catalog index (DDLCATX) area
idms.syssql.ddlcatx	File-ID of the SQL catalog index (DDLCATX) area
asfdml	Filename of the asf dictionary definition (DDLDMML) area
idms.asfdict.ddldml	File-ID of the asf dictionary definition (DDLDMML) area
asflod	Filename of the asf dictionary definition load (ASFLOD) area
idms.asfdict.asflod	File-ID of the asf dictionary definition load (ASFLOD) area
asfdata	Filename of the asf data (ASFDATA) area
idms.asfdict.asfdata	File-ID of the asf data area (ASFDATA) area
ASFDEFN	Filename of the asf data definition (ASFDEFN) area
idms.asfdict.asfdefn	File-ID of the asf data definition area (ASFDEFN) area
j1jrnl	Filename of the first disk journal file
idms.j1jrnl	File-ID of the first disk journal file
j2jrnl	Filename of the second disk journal file
idms.j2jrnl	File-ID of the second disk journal file
j3jrnl	Filename of the third disk journal file
idms.j3jrnl	File-ID of the third disk journal file
sysidms.parms	Filename of the SYSIDMS parameter file

---

# Appendix C. VM/ESA Commands

---

- C.1 Running in local mode . . . . . C-3
  - C.1.1 Central version modifications . . . . . C-5
  - C.1.2 Creating the SYSIPT or SYSIDMS file . . . . . C-5



## C.1 Running in local mode

Commands to execute CA-IDMS reports in **local mode** are shown below. To execute the reports in local mode, take one of the following actions:

- Code LOCAL as an option on the DATABASE parameter. For more information about this option, see the *CA-CULPRIT Reference*.
- Code PARM='\*LOCAL\*' on the OSRUN command used to invoke the program. This option is valid only if the OSRUN command is issued from the System Product Interpreter or an EXEC2 file.
- Link edit the program with an IDMSOPTI module that specifies CENTRAL=NO.

### CULPRIT (VM/ESA)

```
*----- CA-IDMSRPTS -----*
GLOBAL TXTLIB sortlib
FILEDEF SYS004 PRINTER (RECFM FBA LRECL 133 BLKSIZE 133
FILEDEF SYS005 DISK uprmwork file a (RECFM FB LRECL 320 BLKSIZE 1600
FILEDEF CULSRT1I DISK IDMSLIB MACLIB A (MEMBER SORT1
FILEDEF SYS006 DISK uextwork file a (RECFM VB LRECL 1024 BLKSIZE 4096
FILEDEF SYS007 DISK srtpwork file a (RECFM F LRECL 80 BLKSIZE 80
FILEDEF SYS008 DISK nsrtwork file a (RECFM VB LRECL 1024 BLKSIZE 4096
FILEDEF SYS010 DISK input file a
FILEDEF SYS020 DISK nonprint file a
FILEDEF SYSPCH DISK card output a
FILEDEF SYSIPT DISK sysipt data a
FILEDEF SYSIN4 DISK restart file a
FILEDEF CULLIB DISK srclib MACLIB a
FILEDEF sysjrn1 DUMMY
FILEDEF SYSIDMS DISK sysidms parms a (RECFM F LRECL pppp BLKSIZE pppp
EXEC IDMSFD
OSRUN CULPRIT
```

card output a	filename, filetype, and filemode of the file that contains the card output (required for DREPORT 051 and CREPORT 051)
IDMSFD	Name of the exec provided at installation that contains the file definitions for CA-IDMS dictionaries, sample databases, and disk journal files, TXTLIBs, and LOADLIBs
input file a	filename, filetype, and filemode of the primary input file (required for SREPORTs and JREPORTs)
nnnn	number of pages in the file
nonprint file a	filename, filetype, and filemode of the nonprint/nonpunch output (required for SREPORT 099, DREPORT 052 and CREPORT 052; otherwise omitted)
nsrtwork file a	filename, filetype, and filemode of the unsorted extract output data set

---

pppp	page size of the file
restart file a	filename, filetype, and filemode of the file that contains the optional restart parameter; if there is no restart parameter, code DUMMY in place of the filename, filetype, and filemode
sortlib	file name to run external sort package
srclib	file name of PDS containing parameters to be copied (necessary only if USE, =COPY, or =MACRO is used)
srtprwork file a	filename, filetype, and filemode of the sort control parameter file
sysipt data a	filename, filetype, and filemode of the file that contains the CULPRIT parameters for the run  ►► Information about creating this file appears later in this appendix.
sysidms parms a	filename, filetype, and filemode of the SYSIDMS parameters file  ►► Information about creating this file appears later in this appendix.
sysjrn1	file name for the tape journal file, as assigned in the DMCL definition
uextwork file a	filename, filetype, and filemode of the unsorted extracted item data set
uprmwork file a	filename, filetype, and filemode of the unsorted parameter file

---

**Note:** CULPRIT requires an external sort package (other than the VM/ESA SORT command) that can be loaded dynamically.

### C.1.1 Central version modifications

CA-IDMS reports that run under central version can access an IDMS-CV/DC system that is running in a VM/ESA virtual machine. To identify the IDMS-DC/UCF system to be accessed, take one of the following actions:

- Specify `CVMACH=cv-machine-name` on the `DATABASE` parameter, where *cv-machine-name* is a 1- through 8-character user identifier of the VM/ESA virtual machine in which the DC/UCF system is executing. For more information, see the *CA-CULPRIT Reference*.
- Code `PARM='CVMACH=cv-machine-name'` in the `OSRUN` command used to invoke the program. This option is valid only if the `OSRUN` command is issued from the System Product Interpreter or from an `EXEC2` file.
- Link edit the utility with an `IDMSOPTI` module that specifies `CVMACH=cv-machine-name`.

### C.1.2 Creating the SYSIPT or SYSIDMS file

To create the `SYSIPT` file, enter these VM/ESA commands:

```
XEDIT sysipt_data_a (NOPROF
INPUT
.
.
.
Source statements
.
.
.
FILE
```

To create the `SYSIDMS` parameter file, substitute *sysidms parms a* in the example above.

►► For information on all `SYSIDMS` parameters, see *CA-IDMS Database Administration*.





# Appendix D. BS2000/OSD JCL

---

D.1 Job stream ..... D-3



## D.1 Job stream

JCL for executing CA-IDMS reports in either **local mode** or under the **central version** follows:

### CULPRIT (BS2000/OSD)

```
/CALL-PROC (LIB=idms.dba.srclib,ELEM=FILE),PROC-PAR=(MODE=mode)
/ADD-FILE-LINK L-NAME=SYS002,F-NAME=user.keyfile
/ADD-FILE-LINK L-NAME=SYS010,F-NAME=user.inputfil
/ADD-FILE-LINK L-NAME=SYS020,F-NAME=user.nonprint
/CALL-PROC (LIB=idms.dba.srclib,ELEM=CULPRIT),PROC-PAR=(INFP=input.file)
```

idms.dba.srclib	filename of the IDMS dba source library containing the CULPRIT procedure  ►► More information about this file appears in the <i>CA-CULPRIT Reference</i> .
input.file	name of the file which contains the CULPRIT parameters
mode	keyword specifying to run the job either locally (LOCAL) or under the central version (CENTRAL)
user.inputfil	filename of primary input file required for JREPORT and SREPORT runs
user.keyfile	filename of the keyfile, only necessary when a keyfile is input
user.nonprint	filename of the nonprint/nonpunch output, required for SREPORT 099, DREPORT 052, CREPORT 052; omitted otherwise



# Appendix E. CA-IDMS Module Listing

---

E.1	AREPORT listing	E-4
E.2	CREPORT listing	E-5
E.3	DREPORT listing	E-7
E.4	IREPORT listing	E-10
E.5	JREPORT listing	E-11
E.6	QREPORT listing	E-12
E.7	SREPORT listing	E-13



---

This appendix lists all the CA-IDMS AREPORTs, CREPORTs, DREPORTs, IREPORTs, JREPORTs, and SREPORTs in order by module number.

## E.1 AREPORT listing

Table E-1 lists AREPORTs by module number.

Table E-1. CA-ADS Reports — AREPORTs

<b>AREPORT Module</b>	<b>Report Name</b>	<b>KEY Parameter</b>
001	ADS Dialogs and Their Components — Detail	
002	ADS Dialogs and Their Components — Key	KEY PROG-NAME-051 'dialog-name'
003	ADS Dialogs by Process Key	KEY MOD-NAME-067 'process-name'
004	ADS Dialogs by Record Key	KEY RSYN-NAME-079 'record-name'
005	ADS Dialogs by Subschema Key	KEY SS-NAM-026 'subschema-name'
006	ADS Dialogs by Map Key	KEY MAP-NAME-098 'map-name'



## E.2 CREPORT listing

Table E-2 lists CREPORTs by module number.

Table E-2 (Page 1 of 2). Standard IDMS-DC/UCF Reports — CREPORTs

<b>Module Number</b>	<b>Report Name</b>
001	Network Description by Line(1)
002	Network Description by Physical Terminal(1)
003	Network Description by Logical Terminal(1)
004	Program Description(1)
005	Task Description(1)
006	Queue Description(1)
007	Destination Report(1)
011	System Options(1)
014	Network Description by Line(2)
015	Network Description by Physical Terminal(2)
016	Physical Terminals Within Line(2)
017	Network Description by Logical Terminal(2)
018	Logical Terminal by Physical Terminal(2)
019	Program Description(2)
020	Task Description(2)
021	Task Description Within Program(2)
022	Queue Description(2)
023	Queue Description Within Task(2)
024	Destination Report(2)
025	System Options(2)
028	Defined Messages
029	Defined Devices
030	Map Record Indices
031	Map Field Indices
032	Listing of Maps by Panel
033	Listing of Maps

Table E-2 (Page 2 of 2). Standard IDMS-DC/UCF Reports — CREPORTs

<b>Module Number</b>	<b>Report Name</b>
034	Listing of Maps by Record Name
035	Listing of Maps by Element Name
040	ADS/OnLine Report(1)
041	OLQ Report(1)
043	Listing of Nodes
044	Listing of Defined Resources
045	ADS/OnLine Report(2)
046	OLQ Report(2)
050	Load Area Modules
051	Module Text to Card Utility
052	Module Text to Output File Utility
053	Symbol Table Report

(1) Object reports

(2) Source reports

(3) Source and object reports

## E.3 DREPORT listing

Table E-3 lists DREPORTs by module number.

Table E-3 (Page 1 of 3). Standard Data Dictionary Reports — DREPORTs

<b>Module Number</b>	<b>Report Name</b>
00	Housekeeping Module
001	Class Report — Detail
002	Attribute Report — Detail
003	System Report — Detail
004	User Report — Detail
005	Program Report — Detail
006	Module Report — Detail
007	File Report — Detail
008	Record Report — Detail
009	Element Report — Detail
010	Inactive Element Report — Detail
011	Task Report — Detail
012	Queue Report — Detail
013	Destination Report — Detail
014	Logical Terminal Report — Detail
015	Physical Terminal Report — Detail
016	Line Report — Detail
017	Panel Report — Detail
018	Map Report — Detail
019	User-Defined Entity Report — Detail
020	File/Record Cross-Reference Report
021	File Synonym Cross-Reference Report
022	Record Synonym Cross-Reference Report
023	Element Synonym Reference Report
024	Element Description Cross-Reference Report
025	Element Designator Cross-Reference Report

Table E-3 (Page 2 of 3). Standard Data Dictionary Reports — DREPORTs

<b>Module Number</b>	<b>Report Name</b>
026	File Activity Report
027	IDMS Set Activity Report
028	IDMS Record Activity Report
029	IDMS Area Activity Report
030	Element/Program Cross-Reference Report
038	Record/Attribute Report — Key
039	Element/Attribute Report — Detail
050	Level Number Report
051	Module Text to Card Utility
052	Module Text to Output File Utility
053	System Report - Summary
054	User Report - Summary
055	Program Report - Summary
056	Module Report - Summary
057	File Report - Summary
058	Record Report - Summary
059	Element Report - Summary
061	Task Report - Summary
062	Queue Report - Summary
063	Destination Report - Summary
064	Logical Terminal Report - Summary
065	Physical Terminal Report - Summary
066	Line Report - Summary
067	Panel Report - Summary
068	Map Report - Summary
071	Class Report — Key
072	Attribute Report — Key
073	System Report — Key
074	User Report — Key

Table E-3 (Page 3 of 3). Standard Data Dictionary Reports — DREPORTs

<b>Module Number</b>	<b>Report Name</b>
075	Program Report — Key
076	Module Report — Key
077	File Report — Key
078	Record Report — Key
079	Element Report — Key
081	Task Report — Key
082	Queue Report — Key
083	Destination Report — Key
084	Logical Terminal Report — Key
085	Physical Terminal Report — Key
086	Line Report — Key
087	Panel Report — Key
088	Map Report — Key
089	User-Defined Entity Report — Key
090	Catalog Summary Report
091	Catalog Detail Report
092	Group Detail Report
093	User Detail Report
094	Folder Detail Report
095	Object Detail Report
096	Catalog Summary Key Report
097	Catalog Detail Key Report

## E.4 IREPORT listing

Table E-4 lists IREPORTs by module number.

---

Table E-4. ASF Row Level Security Reports — IREPORTs

---

<b>Module Number</b>	<b>Report Name</b>
001	Row Level Security Summary Report
002	Row Level Security Detail Report
003	Row Level Security Summary Report by User
004	Row Level Security Detail Report by User
005	Row Level Security Summary Report by Owner/Security Name

---

---

## E.5 JREPORT listing

Table E-5 lists JREPORTs by module number.

---

Table E-5. Standard Journal Reports — JREPORTs

---

<b>Module Number</b>	<b>Report Name</b>
00	Housekeeping Module
001	Transaction Summary
002	Program Termination Statistics
003	Program I/O Statistics
004	Program Summary
005	Detail Area/Transaction
006	Detail Program/Area
007	Area Summary
008	Formatted Record Dump

---

## E.6 QREPORT listing

Table E-6 lists QREPORTs by module number.

---

Table E-6. SQL Dictionary Reports — QREPORTs

---

<b>Module Number</b>	<b>Report Name</b>
001	SQL Column Name Report
002	SQL Table Information Report
003	SQL Schema Information Report
004	SQL Access Module Information Report
005	SQL Table Access Report
006	SQL Table Syntax Report
007	SQL Table Index Report
008	SQL Table Constraint Report

---



## E.7 SREPORT listing

Table E-7 lists SREPORTs by module number.

Table E-7. Standard Statistics Reports — SREPORTs

<b>Module Number</b>	<b>Report Name</b>
00	Housekeeping Module
001	IDMS Statistics - Histogram Report
003	IDMS-DC System Statistics
005	IDMS-DC Statistics by User Id
006	IDMS-DC Statistics by Lterm Id
007	IDMS-DC Task Statistics by Task Code
008	IDMS-DC ERUS Task Statistics by Accounting Data
009	IDMS-DC ERUS Task Statistics by Program
010	IDMS-DC Transactions Statistics by User Id
011	IDMS-DC Transaction Statistics by Lterm Id
012	IDMS-DC Task Summary
013	IDMS-DC Program Summary
014	IDMS-DC Queue Summary
015	IDMS-DC Line Summary
016	IDMS-DC Physical Terminal Summary
017	Summary of Records Read
018	ADS/OnLine Statistics by User Id
019	ADS/OnLine Statistics by Dialog and Version Number
020	ADS/OnLine Statistics by Logical Terminal Id
021	IDMS-DC Transaction Statistics by Dialog
099	Output File of Archived Statistics Records



# Index

---

## Special Characters

#HSTDS 9-7  
#LGRDS 9-7  
#STLDS 9-7

## A

ABRT journal record 8-7, 8-8  
ADSOBPLG utility 11-7  
ADSORPTS utility 11-7  
AFTR journal record 8-7, 8-8  
archive journal file 8-15, 8-16  
ARCHIVE JOURNAL utility 8-15  
ARCHIVE LOG utility 9-11, 9-14  
area  
    page range 8-40  
    usage mode 8-31, 8-33, 8-34, 8-40  
area activity 2-73—2-74  
AREA journal record 8-6, 8-8  
Area Summary report 8-34  
AREPORT 001 5-8—5-11  
AREPORT 002 5-8—5-11  
AREPORT 003 5-12—5-13  
AREPORT 004 5-14  
AREPORT 005 5-16  
AREPORT 006 5-17  
ASF row-level security reports  
    producing 7-4  
    sample reports 7-6  
    syntax 7-4  
    table of E-10  
attribute reports  
    DREPORT 002 - Detail 2-41—2-43  
    DREPORT 072 - Key 2-41, 2-43

## B

BACKUP utility 11-7  
basic entity 2-6  
basic entity reports 2-12—2-38  
    element reports 2-33—2-38  
    file reports 2-26—2-28  
    module reports 2-23—2-26  
    program reports 2-20—2-23  
    record reports 2-29—2-33  
    system reports 2-12—2-15  
    user reports 2-15—2-19

batch reporting facilities 11-7  
BFOR journal record 8-7, 8-8  
BGIN journal record 8-6, 8-8  
BS2000/OSD JCL D-3  
BUFFER keyword 1-11  
buffer pool, size 8-28  
buffer utilization ratio 8-26, 8-28, 9-24, 9-40  
builder codes 4-57  
BYPASS parameter 1-10

## C

CA-ADS dialog statistics 9-46, 9-52  
CA-ADS reports 5-3—5-17  
    ADS Dialogs and Their Components 5-8—5-11  
    ADS Dialogs by Map Key 5-17  
    ADS Dialogs by Process Key 5-12—5-13  
    ADS Dialogs by Record Key 5-14—5-15  
    ADS Dialogs by Subschema Key 5-16  
    CREPORT 40 4-10  
    CREPORT 45 4-10  
    dialog definition 2-20  
    dialog statistics 9-46—9-52  
    module definition 2-23  
    producing CA-ADS reports 5-6—5-7  
    syntax 5-6—5-7  
    table of 5-4, E-4  
    uses for CA-ADS reports 5-5  
CA-ADS Statistics by Dialog report 9-47  
CA-ADS Statistics by Logical Terminal report 9-48  
CA-ADS Statistics by User report 9-46  
CA-ADS/Batch print log utility  
    *See* ADSOBPLG utility  
CA-CULPRIT  
    coding guidelines 10-6  
    parameter types 10-6  
    parameter types, table 10-9  
CA-ICMS  
    ASF row-level security reports 7-3  
    catalog reports 3-3  
CA-ICMS catalog reports  
    *See* catalog reports  
CA-IDMS reports  
    coding in CA-CULPRIT 10-6  
    example input parameters 1-13  
    general information 1-3  
    permanent modifications 10-16—10-19  
    producing 1-6

- 
- CA-IDMS reports (*continued*)
    - temporary modifications 10-12—10-15
  - CA-IDMS/DB SQL Dictionary Reports 6-3
    - Column Name Report 6-5—6-7
    - producing CA-IDMS/DB SQL Dictionary Reports 6-4
    - Schema and Table Report 6-13—6-16
    - SQL Access Module Information 6-17—6-19
    - SQL Table Access Report 6-21
    - SQL Table Constraint Report 6-28—6-29
    - SQL Table Index Report 6-25—6-27
    - SQL Table Syntax Report 6-22—6-24
    - Table and Column Report 6-8—6-12
    - table of 6-4
  - CA-OLQ 11-4
  - CA-OLQ reports 4-34—4-36
    - CREPORT 041 4-34—4-36
    - CREPORT 046 4-34—4-36
  - CALC cluster ratio 8-23, 8-28, 9-25, 9-40
  - Catalog Detail Key report 3-24
  - Catalog Detail report 3-11
  - catalog reports
    - DREPORT 090 - Catalog Summary 3-9
    - DREPORT 091 - Catalog Detail 3-11—3-13
    - DREPORT 092 - Group Detail 3-14—3-15
    - DREPORT 093 - User Detail 3-16—3-17
    - DREPORT 094 - Folder Detail 3-19—3-20
    - DREPORT 095 - Object Detail 3-21—3-22
    - DREPORT 096 - Catalog Summary Key 3-23
    - DREPORT 097 - Catalog Detail Key 3-24—3-26
    - producing catalog reports 3-7—3-8
    - syntax for 3-7—3-8
    - table of 3-6
    - uses of catalog reports 3-5
  - Catalog Summary Key report 3-23
  - Catalog Summary report 3-9
  - checkpoints
    - See* types of journal records
  - class reports
    - DREPORT 001 - Detail 2-39—2-41
    - DREPORT 071 - Key 2-39, 2-41
  - coding rules
    - See* syntax
  - COMT journal record 8-6, 8-8
  - CREPORT 001 4-29—4-33
  - CREPORT 002 4-29—4-33
  - CREPORT 003 4-29—4-33
  - CREPORT 004 4-37—4-39
  - CREPORT 005 4-53—4-56
  - CREPORT 006 4-40—4-41
  - CREPORT 007 4-16—4-17
  - CREPORT 011 4-43—4-52
  - CREPORT 014 4-29—4-33
  - CREPORT 015 4-29—4-33
  - CREPORT 016 4-29—4-33
  - CREPORT 017 4-29—4-33
  - CREPORT 018 4-29—4-33
  - CREPORT 019 4-37—4-39
  - CREPORT 020 4-53—4-56
  - CREPORT 021 4-53—4-56
  - CREPORT 022 4-40—4-41
  - CREPORT 023 4-40—4-41
  - CREPORT 024 4-16—4-17
  - CREPORT 025 4-43—4-52
  - CREPORT 028 4-14—4-15
  - CREPORT 029 4-13
  - CREPORT 030 4-19—4-22
  - CREPORT 031 4-19—4-22
  - CREPORT 032 4-19—4-22
  - CREPORT 033 4-19—4-22
  - CREPORT 034 4-19—4-22
  - CREPORT 035 4-19—4-22
  - CREPORT 040 4-10—4-12
  - CREPORT 041 4-34—4-36
  - CREPORT 043 4-23—4-24
  - CREPORT 044 4-23—4-24
  - CREPORT 045 4-10—4-12
  - CREPORT 046 4-34—4-36
  - CREPORT 050 4-18
  - CREPORT 051 4-25—4-26
  - CREPORT 052 4-27—4-28
  - CREPORT 053 4-42
  - CREPORTs by number (table) E-5
  - cross-reference reports 2-64—2-75
    - DREPORT 020 - File/Record 2-65
    - DREPORT 021 - File Synonym 2-66
    - DREPORT 022 - Record Synonym 2-66—2-67
    - DREPORT 023 - Element Synonym 2-67—2-68
    - DREPORT 024 - Element Description 2-68—2-69
    - DREPORT 025 - Element Designator 2-70
    - DREPORT 026 - File Activity 2-70—2-71
    - DREPORT 027 - IDMS Set Activity 2-71—2-72
    - DREPORT 028 - IDMS Record Activity 2-72—2-73
    - DREPORT 029 - IDMS Area Activity 2-73—2-74
    - DREPORT 030 - Element/Program 2-74—2-75
  - CULLJRNL module 8-15
- 
- ## D
- database key
    - line number 8-41
-

---

database key (*continued*)  
  page number 8-41  
Database Manipulation Language  
  verb number 8-41  
database pages  
  *See* pages  
DATABASE parameter 1-6—1-7  
DC/UCF Line Summary report 9-31  
DC/UCF Physical Terminal Summary report 9-33  
DC/UCF Program Summary report 9-29  
DC/UCF Queue Summary report 9-31  
DC/UCF run-time statistics 9-3  
DC/UCF system reports  
  CA-ADS parameters 4-10—4-12  
  CA-OLQ 4-34—4-36  
  defined devices 4-13  
  defined messages 4-14—4-15  
  destination 4-16—4-17  
  load area 4-18  
  mapping 4-19—4-22  
  network description 4-29—4-33  
  node and resource table reports 4-23—4-24  
  producing DC/UCF system reports 4-7—4-9  
  program description 4-37—4-39  
  queue description 4-40—4-41  
  syntax for 4-7  
  system options 4-43—4-52  
  table of 4-5  
  task description 4-53—4-56  
  uses of DC/UCF system reports 4-4  
DC/UCF System Statistics report 9-17  
DCMT DISPLAY commands 11-5  
DDDL commands 11-5—11-6  
deadlock prevention element (DPE) 9-20, 9-23  
destination reports 2-50—2-52, 4-16—4-17  
  CREPORT 007 4-16—4-17  
  CREPORT 024 4-16—4-17  
  DREPORT 013 - Detail 2-50—2-52  
  DREPORT 063 - Summary 2-50  
  DREPORT 083 - Key 2-50, 2-52  
Detail Area/Transaction report 8-30  
detail parameter 10-7  
Detail Program/Area report 8-32  
detail reports  
  area/transaction 8-30  
  attribute 2-41—2-43  
  CA-ADS dialog 5-8  
  catalog 3-11—3-13  
  catalog detail key 3-24—3-26  
  class 2-39—2-41  
  destination 2-50—2-52  
  detail reports (*continued*)  
    element 2-34—2-36  
    folder 3-19—3-20  
    general description of 2-7  
    group 3-14—3-15  
    IDD file 2-26—2-28  
    inactive element 2-36—2-37  
    line 2-57—2-58  
    logical terminal 2-52—2-54  
    map 2-61—2-63  
    module 2-24—2-26  
    object 3-21—3-22  
    panel 2-59—2-60  
    program 2-20—2-23  
    program/area 8-32  
    queue 2-48—2-50  
    record 2-29—2-32  
    system 2-13—2-15  
    task 2-46—2-48  
    user 2-16—2-19  
    user catalog 3-16—3-17  
    user-defined entity 2-43—2-44  
  device reports 4-13  
  dictionary builder codes 4-57  
  dictionary report categories 2-6—2-7  
  dictionary reports  
    *See* standard dictionary reports  
  DML  
    *See* Database Manipulation Language  
  DPE  
    *See* deadlock prevention element (DPE)  
  DREPORT 000 2-76  
  DREPORT 001 2-39—2-41  
  DREPORT 002 2-41—2-43  
  DREPORT 003 2-13—2-15  
  DREPORT 004 2-16—2-19  
  DREPORT 005 2-20—2-23  
  DREPORT 006 2-24—2-26  
  DREPORT 007 2-26—2-28  
  DREPORT 008 2-29—2-32  
  DREPORT 009 2-34—2-36  
  DREPORT 010 2-36—2-37  
  DREPORT 011 2-46—2-48  
  DREPORT 012 2-48—2-50  
  DREPORT 013 2-50—2-52  
  DREPORT 014 2-52—2-54  
  DREPORT 016 2-57—2-58  
  DREPORT 017 2-59—2-60  
  DREPORT 018 2-61—2-63  
  DREPORT 019 2-43—2-44

---

---

DREPORT 020 2-65  
DREPORT 022 2-66—2-67  
DREPORT 023 2-67—2-68  
DREPORT 024 2-68—2-69  
DREPORT 025 2-70  
DREPORT 026 2-70—2-71  
DREPORT 027 2-71—2-72  
DREPORT 028 2-72—2-73  
DREPORT 029 2-73—2-74  
DREPORT 030 2-74—2-75  
DREPORT 038 2-32—2-33  
DREPORT 039 2-37—2-38  
DREPORT 050 2-76—2-77  
DREPORT 051 2-77—2-78  
DREPORT 052 2-78—2-79  
DREPORT 053 2-13  
DREPORT 054 2-15—2-16  
DREPORT 055 2-20  
DREPORT 056 2-23  
DREPORT 057 2-26  
DREPORT 058 2-29  
DREPORT 059 2-34  
DREPORT 061 2-46  
DREPORT 062 2-48  
DREPORT 063 2-50  
DREPORT 064 2-52  
DREPORT 065 2-54  
DREPORT 066 2-56—2-57  
DREPORT 067 2-58—2-59  
DREPORT 068 2-61  
DREPORT 071 2-39—2-41  
DREPORT 072 2-41—2-43  
DREPORT 073 2-13—2-15  
DREPORT 074 2-16—2-19  
DREPORT 075 2-20—2-23  
DREPORT 076 2-24—2-26  
DREPORT 077 2-26—2-28  
DREPORT 078 2-29—2-32  
DREPORT 079 2-34—2-36  
DREPORT 081 2-46—2-48  
DREPORT 082 2-48—2-50  
DREPORT 083 2-50—2-52  
DREPORT 084 2-52—2-54  
DREPORT 086 2-57—2-58  
DREPORT 087 2-59—2-60  
DREPORT 088 2-61—2-63  
DREPORT 089 2-43—2-44  
DREPORT 090 3-9  
DREPORT 091 3-11—3-13  
DREPORT 092 3-14—3-15

DREPORT 093 3-16—3-17  
DREPORT 094 3-19—3-20  
DREPORT 095 3-21—3-22  
DREPORT 096 3-23  
DREPORT 097 3-24—3-26  
DSECTs  
    for statistics records 9-7

## E

edit parameters 10-7, 10-8  
edit/code tables  
    *See* module reports  
    *See* program reports  
effectiveness ratio 8-24, 8-28, 9-25, 9-41  
EJECT keyword 1-9  
element description 2-68—2-69  
element designator 2-70  
element reports 2-33—2-38  
    DREPORT 009 - Detail 2-34—2-36  
    DREPORT 010 - Inactive Element 2-36—2-37  
    DREPORT 039 - Attribute/Element 2-37—2-38  
    DREPORT 059 - Summary 2-34  
    DREPORT 079 - Key 2-34—2-36  
element synonyms 2-67—2-68  
element/program relationships 2-74—2-75  
ENDJ journal record 8-7, 8-8  
entry points  
    *See* program reports  
extended architecture 9-21  
external request unit service (ERUS) statistics  
    reports 9-35—9-42

## F

file activity 2-70—2-71  
file reports 2-26—2-28  
    DREPORT 007 - Detail 2-26—2-28  
    DREPORT 057 - Summary 2-26  
    DREPORT 077 - Key 2-26, 2-28  
file synonyms 2-66  
file/record relationships 2-65  
Folder Detail report 3-19  
Formatted Record Dump report 8-36

## G

Group Detail report 3-14

---

## H

heading parameter 10-7

histogram

log record layout 9-9

report 9-53

statistics 9-53

## I

IDD dictionary reports

*See* standard dictionary reports

IDMS-DC ERUS Task Statistics by Accounting Data  
report 9-37

IDMS-DC ERUS Task Statistics by Program Name  
report 9-37

IDMS-DC Task Statistics by Lterm Id report 9-36

IDMS-DC Task Statistics by Task Code report 9-36

IDMS-DC Task Statistics by user Id report 9-35

IDMS-DC Task Summary report 9-27

IDMS/DC Transaction Statistics by Dialog report 9-44

IDMS/DC Transaction Statistics by Lterm Id report 9-43

IDMS/DC Transaction Statistics by User Id report 9-43

IDMSDBAN utility 8-5, 11-7

IDMSDUMP utility 8-5

IDMSLBLS procedure B-5

IDMSRPTS utility 11-7

input file

for statistics reports 9-12

INPUT parameter 10-7

for journal reports 8-16

for statistics reports 9-12

Input Parameter Listing 1-7

internal lock element (ILE) 9-20

IREPORT 001 7-6

IREPORT 002 7-6

IREPORT 003 7-6

IREPORT 004 7-6

IREPORT 005 7-7

IREPORTs by number (table) E-10

## J

JCL

BS2000/OSD D-3

OS/390 A-3—A-6

VM/ESA C-3—C-5

VSE/ESA B-3—B-10

journal file

record sequence number 8-38

journal record entries 8-7

journal records 8-14

ABRT journal record 8-7

AFTR journal record 8-7

AREA journal record 8-6

BFOR journal record 8-7

BGIN journal record 8-6

checkpoints 8-6—8-7

COMT journal record 8-6

ENDJ journal record 8-7

field definitions 8-11

journal record entries 8-7—8-8

record layouts 8-8—8-11

RTSV journal record 8-7

table of 8-14

TIME journal record 8-6

journal reports 8-3—8-41

Area Summary reports 8-34—8-35

Detail Area/Transaction report 8-30—8-31

Detail Program/Area report 8-32—8-33

Formatted Record Dump report 8-36—8-41

producing journal reports 8-15—8-19

Program I/O Statistics report 8-25—8-26

Program Summary report 8-27—8-29

Program Termination Statistics report 8-22—8-24

table of 8-4, E-11

Transaction Summary report 8-20—8-21

uses for journal reports 8-5

JREPORT 000 8-11

JREPORT 001 8-20—8-21

JREPORT 002 8-22—8-24

JREPORT 003 8-25—8-26

JREPORT 004 8-27—8-29

JREPORT 005 8-30—8-31

JREPORT 006 8-32—8-33

JREPORT 007 8-34—8-35

JREPORT 008 8-36—8-41

JREPORTs by number (table) E-11

JSEG dumps 8-36

## K

KEY parameter 1-10, 10-7

in AREPORTs 5-6

in catalog reports 3-7

in DREPORTs 2-8

key reports

*See also* KEY parameter

ADS dialog by map key 5-17

ADS dialog by process key 5-12

ADS dialog by record key 5-14

---

key reports (*continued*)

- ADS dialog by subschema key 5-16
- AREPORT keyfields 5-4
- attribute 2-41—2-43
- attribute/element 2-37—2-38
- CA-ADS dialog 5-8
- catalog detail key 3-24—3-26
- catalog summary key 3-23
- class 2-39—2-41
- destination 2-50—2-52
- element 2-34—2-36
- general description of 2-7
- IDD file 2-26—2-28
- line 2-57—2-58
- logical terminal 2-52—2-54
- map 2-61—2-63
- module 2-24—2-26
- panel 2-59—2-60
- program 2-20—2-23
- queue 2-48—2-50
- record 2-29—2-32
- record/attribute 2-32—2-33
- system 2-13—2-15
- task 2-46—2-48
- user 2-16—2-19
- user-defined entity 2-43—2-44

## L

LANGUAGE 2-39

*See also* site-specific entity reports

level numbers 2-76—2-77

line reports 2-56—2-58

DREPORT 016 - Detail 2-57—2-58

DREPORT 066 - Summary 2-56—2-57

DREPORT 086 - Key 2-57, 2-58

SREPORT 015 9-31

LIST keyword 1-9

load area reports 4-18

locks 8-23, 8-40

log file 9-3

log record 9-9

logical terminal reports 2-52—2-54

DREPORT 014 - Detail 2-52—2-54

DREPORT 064 - Summary 2-52

DREPORT 084 - Key 2-52, 2-54

## M

map reports 2-60—2-63

AREPORT 005 5-16

map reports (*continued*)

CREPORT 030 4-19—4-22

CREPORT 031 4-19—4-22

CREPORT 032 4-19—4-22

CREPORT 033 4-19—4-22

CREPORT 034 4-19—4-22

CREPORT 035 4-19—4-22

DREPORT 018 - Detail 2-61—2-63

DREPORT 068 - Summary 2-61

DREPORT 088 - Key 2-61, 2-63

mapping batch utility

*See* RHDCMPUT utility

message reports 4-14—4-15

MODE 2-39

*See also* site-specific entity reports

module reports 2-23—2-26

DREPORT 006 - Detail 2-24—2-26

DREPORT 056 - Summary 2-23

DREPORT 076 - Key 2-24, 2-26

## N

network description reports 4-29—4-33

CREPORT 001 4-29—4-33

CREPORT 002 4-29—4-33

CREPORT 003 4-29—4-33

CREPORT 014 4-29—4-33

CREPORT 015 4-29—4-33

CREPORT 016 4-29—4-33

CREPORT 017 4-29—4-33

CREPORT 018 4-29—4-33

node and resource table reports

CREPORT 043 4-23—4-24

CREPORT 044 4-23—4-24

NOLIST keyword 1-9

## O

Object Detail report 3-21

OLP 11-4—11-5

online messages 4-14—4-15

OnLine PLOG

*See* OLP

Online Query

*See* CA-OLQ

online reporting facilities

CA-OLQ 11-4

DCMT commands 11-5

DDDL commands 11-5—11-6

OLP 11-4—11-5

OPER commands 11-5



---

OPER commands 9-21  
    program pool statistics 9-21  
OPER WATCH commands 11-5  
OS/390 JCL A-3—A-6  
OUTPUT parameter 10-7

## P

pages  
    database page displacement 8-41  
    read from database 8-26, 8-27, 8-39, 9-25  
    requested from database 8-26, 8-39, 9-24, 9-40  
    written to database 8-26, 8-28, 8-39, 9-25, 9-40  
panel reports 2-58—2-60  
    DREPORT 017 - Detail 2-59—2-60  
    DREPORT 067 - Summary 2-58—2-59  
    DREPORT 087 - Key 2-59—2-60  
PARAM= parameter 1-7—1-9  
PATH parameter 1-9, 1-11, 10-7  
physical terminal reports 2-54—2-56  
    DREPORT 015 - Detail 2-55  
    DREPORT 065 - Summary 2-54  
    DREPORT 085 - Key 2-55  
    SREPORT 016 9-33  
print control requests 1-9  
PRINT INDEX utility 11-7  
PRINT JOURNAL utility 11-7  
PRINT LOG utility 11-7  
PRINT PAGE utility 11-7  
PRINT SPACE utility 11-7  
printing report modules 2-77—2-79, 4-25—4-28  
process parameters 10-7  
Program I/O Statistics report 8-25  
program pools 9-21  
    extended architecture 9-21  
    program pages 9-21  
    reentrant 9-21  
    waits 9-21  
program reports 2-20—2-23  
    CREPORT 004 4-37—4-39  
    CREPORT 019 4-37—4-39  
    DREPORT 005 - Detail 2-20—2-23  
    DREPORT 055 - Summary 2-20  
    DREPORT 075 - Key 2-20, 2-23  
    JREPORT 002 8-22  
    JREPORT 003 8-25  
    SREPORT 013 9-29  
Program Summary report 8-27  
Program Termination Statistics report 8-22  
programs  
    loads 9-21

programs (*continued*)  
    pages 9-21  
    program pool waits 9-21  
    reentrant 9-21  
    times run 8-27  
punching report modules 2-77—2-79, 4-25—4-28

## Q

q-files  
    *See* module reports  
QREPORT 001 6-5  
QREPORT 002 6-8  
QREPORT 003 6-13  
QREPORT 004 6-17  
QREPORT 005 6-20  
QREPORT 006 6-22  
QREPORT 007 6-25  
QREPORT 008 6-28  
QREPORTS  
    *See* CA-IDMS/DB SQL Dictionary Reports  
QREPORTs by number (table) E-12  
queue reports 2-48—2-50  
    CREPORT 006 4-40—4-41  
    CREPORT 022 4-40—4-41  
    CREPORT 023 4-40—4-41  
    DREPORT 012 - Detail 2-48—2-50  
    DREPORT 062 - Summary 2-48  
    DREPORT 082 - Key 2-48, 2-50  
    SREPORT 014 9-31

## R

RCE  
    *See* resource control element (RCE)  
RDW  
    *See* record descriptor word  
REC parameters 10-7  
    for journal reports 8-11  
    for statistics reports 9-10  
record activity 2-72—2-73  
record reports  
    DREPORT 008 - Detail 2-29—2-32  
    DREPORT 038 - Attribute Key 2-32—2-33  
    DREPORT 058 - Summary 2-29  
    DREPORT 078 - Key 2-29, 2-32  
record summary statistics report 9-57—9-60  
    *See also* statistics reports  
record synonyms 2-66—2-67  
record/element associations  
    *See* record reports

---

- records 8-23
  - CALC 8-23, 8-39
  - current of transaction 8-23, 8-39, 9-25
  - database key 8-41
  - database record display 8-41
  - length for statistics reports 9-12
  - relocated 8-24, 8-40, 9-26, 9-41
  - requested from database 8-23, 8-28, 8-39, 9-25
  - updated 8-39
  - user record 8-41
  - variable length 8-24, 8-40, 9-26, 9-41
  - VIA 8-23, 8-40
- reentrant pools 9-21
  - See also* program pools
- report listings 1-7
- REPORT= parameter 1-9—1-10
- resource control element (RCE) 9-20, 9-23
- resource link element (RLE) 9-19, 9-23
- RHDCMPUT utility 11-7
- RLE
  - See* resource link element (RLE)
- ROLLBACK utility 11-7
- ROLLFORWARD utility 11-7
- Row Level Security Detail By User report 7-6
- Row Level Security Detail report 7-6
- Row Level Security Summary by Owner/Security Name report 7-7
- Row Level Security Summary By User report 7-6
- Row Level Security Summary report 7-6
- RTSV journal record 8-7, 8-8

## S

- SELECT and BYPASS parameters 1-10—1-13, 10-7
  - BUFFER keyword 1-11
  - for journal reports 8-16
- SELECT parameter 1-10
- Sequential Parameter Listing 1-7
- session indicator
  - See* statistics reports
- set activity 2-71—2-72
- site-specific entity reports
  - attribute reports 2-41—2-43
  - class reports 2-39—2-41
  - user-defined entity reports 2-43—2-44
- SORT parameter 10-7
- space management ratio 8-29, 9-25, 9-41
- special-purpose report modules
  - CREPORT 051 - Module to Card Utility 4-25
  - DREPORT 000 - Comment/Nest Resolution 2-76
  - DREPORT 050 - Level Number 2-76—2-77

- special-purpose report modules (*continued*)
  - DREPORT 051 - Module to Card Utility 2-77—2-78
  - DREPORT 052 - Module to File Utility 2-78—2-79
- SREPORT 000 9-10, 9-12
- SREPORT 001 9-12, 9-53—9-56
- SREPORT 003 9-12, 9-17—9-27
- SREPORT 005 9-35, 9-38, 9-42
- SREPORT 006 9-36, 9-38, 9-42
- SREPORT 007 9-36, 9-38, 9-42
- SREPORT 008 9-37, 9-38, 9-42
- SREPORT 009 9-37, 9-38, 9-42
- SREPORT 010 9-43
- SREPORT 011 9-43
- SREPORT 012 9-12, 9-27—9-28
- SREPORT 013 9-12, 9-29—9-30
- SREPORT 014 9-12, 9-31
- SREPORT 015 9-12, 9-31—9-32
- SREPORT 016 9-12, 9-33—9-34
- SREPORT 017 9-57—9-60
- SREPORT 018 9-46, 9-49—9-52
- SREPORT 019 9-47, 9-49—9-52
- SREPORT 020 9-48, 9-49—9-52
- SREPORT 021 9-44
- SREPORT 099 9-10, 9-16
- SREPORTs by number (table) E-13
- standard dictionary reports
  - basic entities 2-12
  - basic entity reports 2-12—2-38
  - catalog reports 3-3—3-26
  - categories 2-6—2-7
  - cross-reference 2-64—2-75
  - producing dictionary reports 2-8—2-11
  - site-specific entity reports 2-39—2-44
  - special-purpose report modules 2-76—2-79
  - table of E-7
  - teleprocessing entity reports 2-45—2-63
  - uses for 2-5
- startup record 9-60
- statistics log records
  - histogram 9-9
  - layout 9-7
  - number of 9-10
  - release level 9-10
  - system statistics 9-9
  - task statistics 9-8
  - transaction statistics 9-8
- statistics reports
  - CA-ADS dialog statistics 9-46—9-52
  - CA-ADS statistics by dialog 9-47
  - CA-ADS statistics by logical terminal 9-48
  - CA-ADS statistics by user 9-46

---

statistics reports (*continued*)

- ERUS statistics by accounting data 9-37
- ERUS statistics by program name 9-37
- ERUS statistics reports 9-35—9-42
- examples 9-13—9-14
- histogram report 9-53—9-56
- line summary 9-31—9-32
- physical terminal summary 9-33—9-34
- producing statistics reports 9-11—9-16
- program summary 9-29—9-30
- queue summary 9-31
- record summary statistics report 9-57—9-60
- session indicator 9-13
- syntax 9-11
- system statistics reports 9-17—9-34
- system wide statistics 9-17—9-27
- table of 9-4
- task statistics by lterm id 9-36
- task statistics by task code 9-36
- task statistics by user id 9-35
- task statistics reports 9-35—9-42
- task summary 9-27—9-28
- transaction statistics by dialog 9-44
- transaction statistics by lterm id 9-43
- transaction statistics by user id 9-43
- transaction statistics reports 9-43—9-45
- types of statistics records 9-7—9-10
- uses for statistics reports 9-5—9-6

storage pools 9-19, 9-20, 9-23

subschemas

- See* program reports

summary reports

- area 8-34
- catalog 3-9
- catalog summary key 3-23
- destination 2-50
- element 2-34
- IDD file 2-26
- line 2-56—2-57, 9-31
- logical terminal 2-52
- map 2-61
- module 2-23
- panel 2-58—2-59
- physical terminal 2-54, 9-33
- program 2-20, 9-29
- queue 2-48, 9-31
- record 2-29
- system 2-13
- task 2-46, 9-27
- user 2-15—2-16

symbol table report 4-42

synonyms

- element 2-67
- file 2-66
- record 2-66

syntax

- for AREPORTs 5-6
- for CA-ADS reports 5-7
- for CA-CULPRIT parameters 10-6
- for CA-IDMS reports 1-6
- for CREPORTs 4-7
- for DREPORTs (catalog) 3-7
- for DREPORTs (dictionary) 2-8
- for IREPORTs 7-4
- for JREPORTs 8-15—8-17
- for SREPORTs 9-11

system log file

- ARCHIVE LOG utility 9-11
- generating 9-11
- writing system statistics to 9-17

system options reports

- CREPORT 011 4-43—4-52
- CREPORT 025 4-43—4-52

system reports

- DREPORT 003 - Detail 2-13—2-15
- DREPORT 053 - Summary Report 2-13
- DREPORT 073 - Key 2-13, 2-15
- table of E-5

system statistics

- log record layout 9-9

system statistics reports

- See also* statistics reports
- table of E-12, E-13

## T

task reports

- CREPORT 005 4-53—4-56
- CREPORT 020 4-53—4-56
- CREPORT 021 4-53—4-56
- DREPORT 011 - Detail 2-46—2-48
- DREPORT 061 - Summary 2-46
- DREPORT 081 - Key 2-46, 2-48
- SREPORT 012 9-27
- statistics 9-35

task, definition 9-28

tasks

- abended 9-19
- active 9-19
- maximum number 9-19
- statistics collection 9-35

---

teleprocessing entity 2-6  
    general description of 2-7

teleprocessing entity reports 2-45—2-63  
    destination reports 2-50—2-52  
    lines 2-56—2-58  
    logical terminal reports 2-52—2-54  
    map reports 2-60—2-63  
    panel reports 2-58—2-60  
    physical terminal reports 2-54—2-56  
    queue reports 2-48—2-50  
    task reports 2-46—2-48

temporary modifications  
    changing headings 10-13  
    copying the report module 10-12  
    deleting columns 10-12—10-13  
    moving columns 10-14—10-15  
    selecting specific data 10-14  
    specifying a new sort sequence 10-14

time  
    in system mode 9-23, 9-40  
    in user mode 9-22, 9-40

TIME journal record 8-6, 8-8

title parameter 10-7

total parameter 10-7

transaction statistics reports 9-43—9-45  
    *See also* statistics reports  
    statistics collection 9-43

Transaction Summary report 8-20

type 0 parameter 10-7

type 1 parameter 10-7

type 3 parameter 10-7

type 4 parameter 10-7

type 5 parameter 10-7

type 6 parameter 10-7

type 7 parameter 10-7

type 8 parameter 10-7

## U

USE parameter 9-13, 10-6

User Detail report 3-16

user reports  
    DREPORT 004 - Detail 2-16—2-19  
    DREPORT 054 - Summary 2-15—2-16  
    DREPORT 074 - Key 2-16, 2-19

user-defined entity reports  
    DREPORT 019 - Detail 2-43—2-44  
    DREPORT 089 - Key 2-43, 2-44

uses for CA-ADS reports 5-5

uses for journal reports 8-5

uses for statistics reports 9-5  
    *See also* statistics reports

uses of catalog reports 3-5

uses of DC/UCF system reports 4-4

utilities  
    *See* batch reporting facilities

## V

variable-length records 9-26

verb number 8-41

VIA cluster ratio 8-23, 8-29, 9-26, 9-41

VM/ESA JCL C-3—C-5

VSE/ESA JCL B-3—B-10

## W

work field parameters 10-7

## X

XA  
    *See* extended architecture



